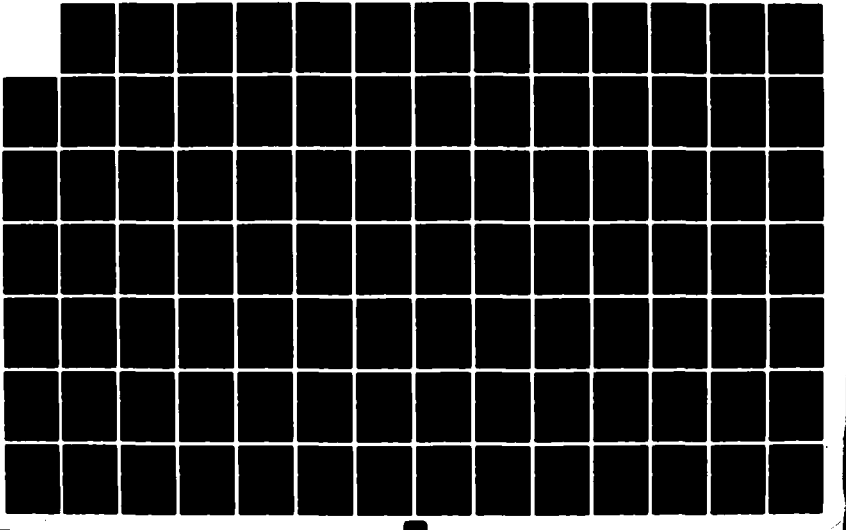
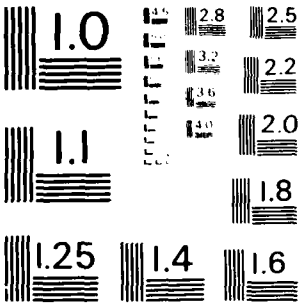
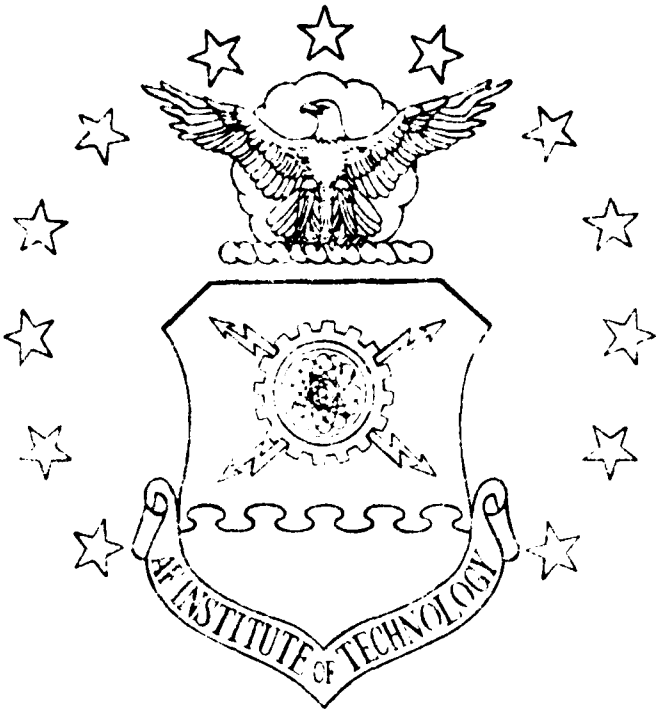
AD-A134 386 COMPUTER ASSISTED INSTRUCTION IN BASIC(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS D J CREAGAN 28 SEP 83 AFIT-LSSR-29-83 1/6

UNCLASSIFIED F/G 9/2 NL







COMPUTER ASSISTED INSTRUCTION IN BASIC

Captain Danny J. Creagan, USAF

LSSR 29-83

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

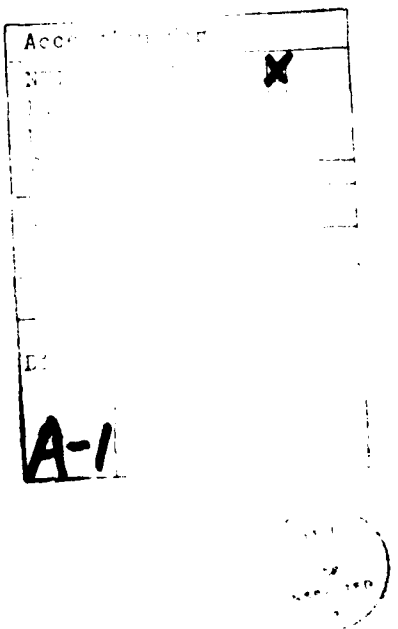
2

COMPUTER ASSISTED INSTRUCTION IN BASIC

Captain Danny J. Creagan, USAF

LSSR 29-83

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information are contained therein. Furthermore, the views expressed in the document are those of the author(s) and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the Air Training Command, the United States Air Force, or the Department of Defense.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER LSSR 29-83	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER ASSISTED INSTRUCTION IN BASIC		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Danny J. Creagan, Captain, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Systems and Logistics Air Force Institute of Technology, WPAFB OH		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Department of Communication AFIT/LSH, WPAFB OH 45433		12. REPORT DATE 28 September 1983
		13. NUMBER OF PAGES 510
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release: IAW AFR 190-17.

LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

15SEP 1983

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computers
Programmed Instruction
Computer Aided Instruction
Programming Languages
Microcomputers

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Thesis Chairman: Jeffrey C. Daneman, GS-13, AFIT/LSQ

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

The Air Force recently approved the purchase of 1500 Cromemco microcomputers. After approximately 400 were installed, various agencies realized that they needed to develop an on-going program to keep their people trained on the machines. However, additional personnel were not authorized to set up a training program. A survey of microcomputer focal points revealed that a Computer Assisted Instruction (CAI) package that taught Beginner's All Purpose Instruction Code (BASIC), would be a good start in developing a training program that required no additional manpower. The author developed a CAI package to satisfy this need. The CAI code was written so that it would conform to Air Force CAI standards, and run on a variety of microcomputers. It was meant to provide sufficient instruction to transform a non-programmer into a programmer competent enough to write mission related programs. It consists of 30 programs, including six lessons (each with two parts), six tests, several homework problems, and a training report maker.

LSSR 29-83

COMPUTER ASSISTED INSTRUCTION IN BASIC

A Thesis

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Logistics Management

By

Danny J. Creagan, BA
Captain, USAF

September 1983

This thesis, written by

Capt Danny J. Creagan

and approved, has been accepted by the undersigned on behalf of the faculty of the School of Systems and Logistics in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS MANAGEMENT

DATE: 28 September 1983

COMMITTEE CHAIRMAN

READER

TABLE OF CONTENTS

CHAPTER 1

BACKGROUND

COMPUTER ASSISTED INSTRUCTION IN BASIC

ISSUE

The Tactical Air Forces recently decided to purchase 1500 Cromemco microcomputers for use at the wing level (Coward, Cokerly and others,82). Headquarters Air Training Command (HQ ATC) will soon follow suit with another small-computer buy (Weber,83). The TAF purchase was based, in part, on the recommendations from a feasibility study by Automated Data Processing personnel from the Tactical Air Command (Coward, 82). The study had been initiated because of numerous reports that local agencies had purchased small computers and successfully used them to do word processing, aircrew flight planning, and simple data base management duties (Carter and Nunley, 81; Coward,82). The study investigated the feasibility of using microcomputers to automate some of the more time consuming tasks commonly encountered in the wings. Concern about non-standardization, security, and duplication of effort when the computers were installed, led the different commands to form small computer offices as focal points for the distribution and control of the computers. The

requirement for receipt of one of the computers was a submission of a OAR (Data Automation Requirement) by the unit (the OAR, in this case, was a formal request which outlined a specific purpose for the computer), and approval of the OAR by the appropriate MAJCOM. Software support was limited to off-the-shelf packages. All of the computers were purchased with Beginners All Purpose Instruction Code (BASIC), a word processor and a data base management program. After purchase, the computers were considered stand alone systems and no increase in manpower to support them was authorized (TACR 300-12, SAC 300-4).

Since the decision to purchase the units, approximately 400 have been installed at a cost of 000 to 20,000 dollars each. The cost was determined by number of peripherals (printers, plotters, disk drives, etc.) that were requested with computers (Coward,82). The expected workload for the computers is light and the machines will be available for other mission-related tasks if the users are able to create some of their own software (Lyon, 83).

PROBLEM

The wings do not have enough programmers available to program the computers, and the programmers who have been identified are not dedicated to the system, so they will

not be replaced when they leave for other assignments. Most personnel, though very interested in the new machines, are not familiar enough with programming techniques to produce adequate software (Cokerly, 82). Thus, when the programmers leave, expertise to write, maintain, and update mission related programs will be lost. The commands need to establish a program to train users in one of the programming languages. The training program needs to be self supporting, and require no additional manpower. It must familiarize new personnel with the computer and its capabilities, and give a non-programmer a solid core of general knowledge in a programming language (Cokerly, 83; Lyon,83).

The BASIC programming language is the best choice for training new operators how to program because BASIC is available on virtually every microcomputer (the Cromemcos purchased by the Air Force included a dialect of BASIC called Microsoft BASIC). Another reason for BASIC is that many good, mission-related programs have already been written in it, showing the language's capability to quickly do time consuming tasks (Carter, Nunley, 81). Additionally, the popularity, usefulness, and ease of modification of BASIC programs has caused the senior members of the Air Force to approve BASIC as an accepted programming language (HQ USAF Message, 82). These reasons have led to a

consensus by HQ TAC personnel that a CAI (Computer Assisted Instruction) program in the BASIC language would provide the most effective solution for the microcomputer training problem. The best dialect of BASIC to use would be the Microsoft BASIC dialect because it is available on the majority of microcomputers (Burke, 82; Blackwood, 82; Lewis, 82; Welch and others, 82).

Colonel Lyon, TAC DOZ, formally outlined the problem in his letter dated 7 February, 1983. He explained the problem with keeping new personnel trained on the computer systems, and suggested that a thesis effort directed towards developing a CAI program in Microsoft BASIC for the newly purchased Cromemco computers could satisfy the needs of his personnel and be cost effective for the Air Force. During a telephone interview, he also stated that commercial programs were not a good choice. A commercial package was hard for each unit to purchase (each software purchase has to go through the formal DAR process), and the programs could not be distributed to other agencies because of copyright restrictions. Colonel Lyon felt that a program developed specifically by the Air Force would be more likely to include those items which the Air Force considers important, and, the software could be distributed without concern for copyright infringement, royalties, etc.

A review of current CAI in BASIC revealed limited

resources adaptable to the problem. The only commercial package available was the Tandy Level II BASIC instruction package. It was not a CAI program (it was not advertised to be one) and it was machine dependent (it was made for the TRS-80 system). It required little feedback from the responder, did not give adequate examples, and made no provision for scoring or reporting on a student's progress. It was a compendium of definitions of computer terms that were intended to be read by the user on a video terminal. This program was too limited to be used as a continuing training program.

The focal point offices for microcomputers were interviewed to find out whether other agencies were developing a CAI. Lt Coward, HQ TAC ADMUDS, was not aware of any related research. Major Cokerly, Air Staff SO-I, also confirmed that a CAI was needed, but he was not aware of any available in the field. Programmers at the offices for computer development and training systems for the Air Force, at Keesler AFB, Mississippi, have developed a CAI for BASIC on the Honeywell computer. However, their program is very machine dependent (it is written in assembly language) and could not be transferred to microcomputers. Searches with other sources, revealed similar need and interest in a CAI program, but none available.

RESEARCH QUESTION

How can a Computer Assisted Instruction program be developed for the new Cromemco microcomputers that provides sufficient instruction to transform a non-programmer into a programmer competent enough to write mission related programs? The CAI would have to be simple enough to adapt to different Cromemco systems, be updated easily (or customized as necessary), yet thorough enough to cover the crucial elements of BASIC, and still be appropriate for an audience with little training in computer concepts. The problem can be broken down into three steps.

The first step would be research of the current literature on both general, and BASIC programming language CAI. Included would be those techniques which are compatible with a microcomputer system and appropriate for the intended military audience.

The second step would be the development of the "frames" for the program. A CAI frame is a portion of a CAI program that contains a single learning event. The events are typically broken into information, question, student response, and feedback/answer sentences (Meredith,71; Freedman,81). A microcomputer usually has limited memory and storage space. The Cromemco machines, when BASIC is implemented, have around 33 kilobytes (33k) of memory. In

addition, the off line storage of microcomputers is sometimes limited to one or two small floppy disk drives (Coward, 82). Therefore, an exhaustive instruction in the language would not be possible, because there would not be enough room for the program in the computer. The frames would have to be isolated into main topics which would sufficiently instruct the student in the language, yet still fit within the storage space and memory limitations of a one disk, 33k machine.

The third step would involve the coding and debugging of the CAI program itself. While the frames should be simple to code, there will have to be hundreds of them to adequately cover the subject. Such a large program will need careful debugging to eliminate errors. Consultation with non-programmers to ensure "user friendliness", and using their feedback to fine tune the CAI, would conclude the research and development of the CAI program.

CHAPTER 2

METHODOLOGY

THE CAI TECHNIQUES ON A MICROCOMPUTER

> When the designer of instructional materials confronts the task of preparing a CAI program, he places instructional design in a new context, that of computer programming. (Holtzman, 70)

Computer assisted programming has much in common with other programmed instruction (PI) methods. However, it has new perspectives and limitations which increases both the student's learning and the instructor's work. Silvern, Holtzman, and other educators have developed outlines for approaching a CAI problem. In their instructions on how to use these outlines, they emphasize that one hour of interactive computer assisted instruction can take the place of many hours of classroom time. Additionally, one hour of CAI may take hundreds of hours of programming and development by teams of instructors, researchers, and programmers. In this chapter, I will develop a methodology for making a CAI package for the BASIC language. Because I do not have the research time needed to develop new techniques, I will concentrate on using proven methods which can be used on a microcomputer.

The Air Force has been interested in how to use CAI

effectively since the early 1970s. They formed an office at Keesler AFB in Mississippi (the CDTS section of the 3300 Technical Training Wing) dedicated to finding, developing and distributing different CAI programs (Ashby, 83). This office has developed guidelines on proper CAI methods to use in their Air Force CAI projects. I have used these guidelines to develop my CAI program.

In addition to the CDTS procedures, the TAC small computer office has published a guide on programming standards for microcomputers for TAC. This unique guide is a first attempt to standardize BASIC programming. It applies to this paper because the standards it outlines, as well as the CDTS standards, must be considered to ensure the finished CAI program satisfies the constraints of all agencies that may wish to use it.

The University of Alaska has donated a CAI package on BASIC to the Air Force (Carew,81). Developed on a mainframe using a special purpose instructional language, the BASIC dialect it teaches is not compatible with Microsoft BASIC. It references devices and equipment not available with a microcomputer. However, the outline it uses is readily adaptable to my project. By using it as guide, but substituting my own text and test items, I have avoided the lengthy process of developing and validating a new approach to CAI in BASIC.

To develop the methodology I have considered the guidelines to effective CAI , as outlined by Holtzman and others, applied this to the standards developed by Keesler CDTS and TAC Small Computer Office, and, using an outline based on the University of Alaska program, modified the result as needed to fit on a microcomputer.

Educators have developed models for creating CAI programs (Holtzman and Silvern,70). These models generally break down the design of CAI into five major areas of consideration. These areas are as follows:

The CAI author must consider the

1. Intent and justification of the CAI
   a. Needs and goals of the CAI
   b. Target audience
   c. Institutional constraints

2. Physical Constraints on Design of CAI
   a. Operational/equipment constraints
   b. Installation constraints

3. Development of the CAI draft
   a. Frame design
   b. Questioning techniques
   c. Scoring of student responses

4. Coding of the CAI program

5. Validation.

An elaboration of each of these areas and how they apply to this paper follows.

Needs and Goals Determination. The needs and goals of the CAI program were discussed in the previous chapter. In summary, the Air Force needs a computer assisted instruction program which teaches the BASIC programming language.

Student Population Description. The target audience is composed of members who are at least high school graduates. Because they oluntarily use the computer to ease their workload, they can be considered motivated to learn the primary programming language of the computer (Lyon, 83).

Institutional Constraints. The biggest institutional constraint is the time needed to develop the program. The MAJCOMs have had to wait several months for the final product. This was an unavoidable constraint. However, the minimal costs of the package have made the delay acceptable.

Physical Constraints on Design of CAI

Operational/equipment. The program library was developed so that it requires only one floppy disk for offline storage. Therefore, any system that has at least one disk drive, and one terminal should be able to install the package. Most systems which support Microsoft BASIC

have this configuration as a minimum (Cokerly,83). Because the program is primarily concerned with BASIC and not with the type of equipment used, it is written so it is compatible with one other popular microcomputer, the TRS-80. (the program will run on the Apple if it is configured with a Z80 CPU card - see the Apple operating manuals for further explanation of how to run Microsoft BASIC on that equipment). This will enables the package to run on the microcomputers which make up the majority of microcomputer installations (Cokerly, 83; Coward, 82; Greene, 82).

The types of terminals and floppy disks vary with brand names of equipment, so a program cannot be easily transferred from one brand name to another, even if the program is in the same language dialect. For instance, Cromemco floppy disks cannot be used in a system that only supports Apple disk operating systems. Therefore, the files had to be transferred using a medium that is common to most computers. The easiest and most common method for transferring programs to dissimilar equipment, is to send them via telephone modem hookups. The CAI package was developed on a TRS-80 system, and transferred to the Cromemco system.

Installation Constraints. The installation constraints deal mainly with the distribution of the software, and

periodic review to ensure it is current. The program will be given to the major command microcomputer focal points for inclusion in their software libraries. Distribution and updates will be the responsibility of the focal point offices in cooperation with the different users. The program will be written in accordance with the TAC guidelines so that it can be updated as needed by any competent BASIC programmer.

Development of the CAI Draft

Frames. CAI and PI experts agree that any CAI should be flowcharted prior to development. This provides a systematic outline for the programmer/author and alleviates many conceptual problems early in program development. After outlining, each major block is broken down into frames. A frame is a complete question and answer sequence, or a complete teaching point (Lysaught, 63; Meridith, 71; Burke, 82). It has the answer to every test item, and it indicates where transfer of the program goes for the different response possibilities of the question.

Questioning techniques (test items). Almost any type of test item can be used in CAI. The limiting factors are the amount of memory required to store the possible test answers (Holtzman and Silvern,70). It would be obvious from this, that essay answers would be very demanding of memory,

and multiple choice, matching, true-false, or one word responses would be more memory conservative (Burke,82). Because of the equipment limitations discussed earlier, I have not used essay questions.

Scoring. The program must provide for record keeping of the scores of different students, especially if access to the system is contingent upon successful completion of training - as may be the case in some organizations. The program uses a simple, sequential file update system, which scores the student after completion of each block of questions. A separate utility program prints out a record of all student scores.

Coding of the CAI Program

The program is too large to fit into memory all at once, so the program must include features which keep track of where the student is, where he wants to go, what his score is, and what module of the program to load next. The student may progress through the program at his own pace and in any order he chooses.

Validation

The CAI program was validated at several stages during its development. As each block was completed, it was tested at AFIT. The purpose of the initial testing was to

eliminate logic errors in the CAI program itself. Therefore, the target audience during the debug phase was knowledgeable in what makes a good program. The program will be sent to the focal point offices and released for testing within the commands. Feedback from these offices will be used to revise the programs as needed.

CHAPTER 3

CAI PROGRAM FORMAT

PROGRAM LIBRARY

There are 30 programs that make up the CAI. There are six lessons (each lesson has two parts), six tests, six score files, four homework assignments, one menu, and one training report maker. The programs require about 350 kilobytes of disk space, and they fit on three data disks for the TRS-80 or on two double sided, double density Cromemco disks. The rest of this chapter describes their purpose, their flowcharts, the subjects they teach, and their limitations.

Program Purpose

	Program Name	Purpose
1.	Lessons 1, 1a, and test 1	Teach Introduction to BASIC and Computer terms
2.	Lessons 2, 2a, and test 2	Teach saving programs to disk, REMARKS, and branching.
3.	Lessons 3, 3a, and test 3	Teach Loops, Arrays, and DIM statements
4.	Lessons 4, 4a, and test 4	Teach Printer commands & Sequential file I/O

5. Lessons 5, 5a, and test 5 ..... Teach Subroutines and Library Functions

6. Lessons 6, 6a, and test 6 ..... Teach String Functions, & Microsoft Editor

7. Menu ..... Lets student select sequence of lessons

8. Training Report program ..... Makes a hard copy report of student scores

9. Score files 1 through 6 ..... Hold student scores

10. Homework assignments 2 thru 5 .... Gives student extra practice in techniques of lesson.

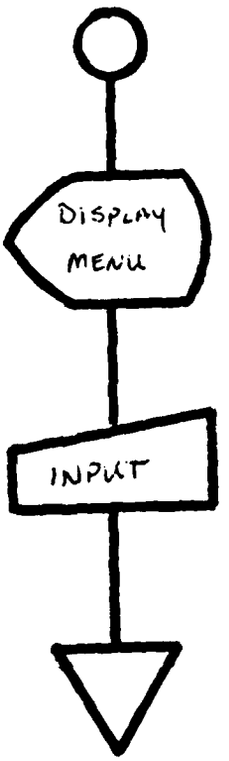
Flowcharts

Lessons

The flowcharts for the lessons are all the same. Each section of a lesson is a subroutine which is called by the lesson menu. The student selects whether to take each section in fixed order, or review selections in the order of his/her choosing. If the lesson is taken in fixed order, the subroutines are called one after another without showing the menu. If the sections are reviewed according to the student's desires, then the menu is displayed after each section is completed. In both cases, the student has the option to continue to the next part or review the section just seen. The following flowchart represents the

main menu program, the one after it represents lessons one through six:

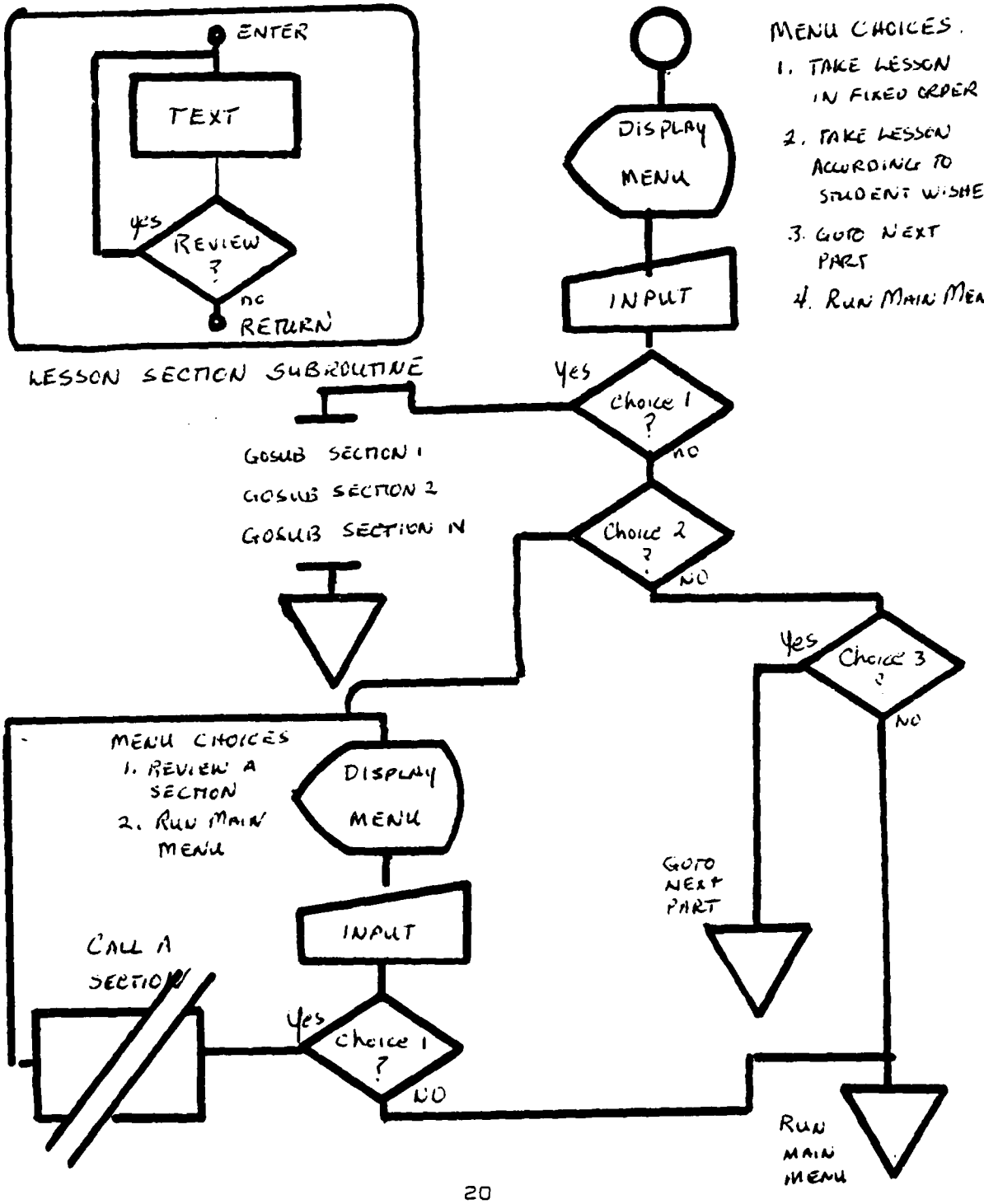
Flowchart For Main Menu



MENU CHOICES.
1. LESSONS ONE - SIX
2. TESTS ONE - SIX

ON CHOICE:
RUN PROGRAM SELECTION

Flowcharts for Lessons One Through Six

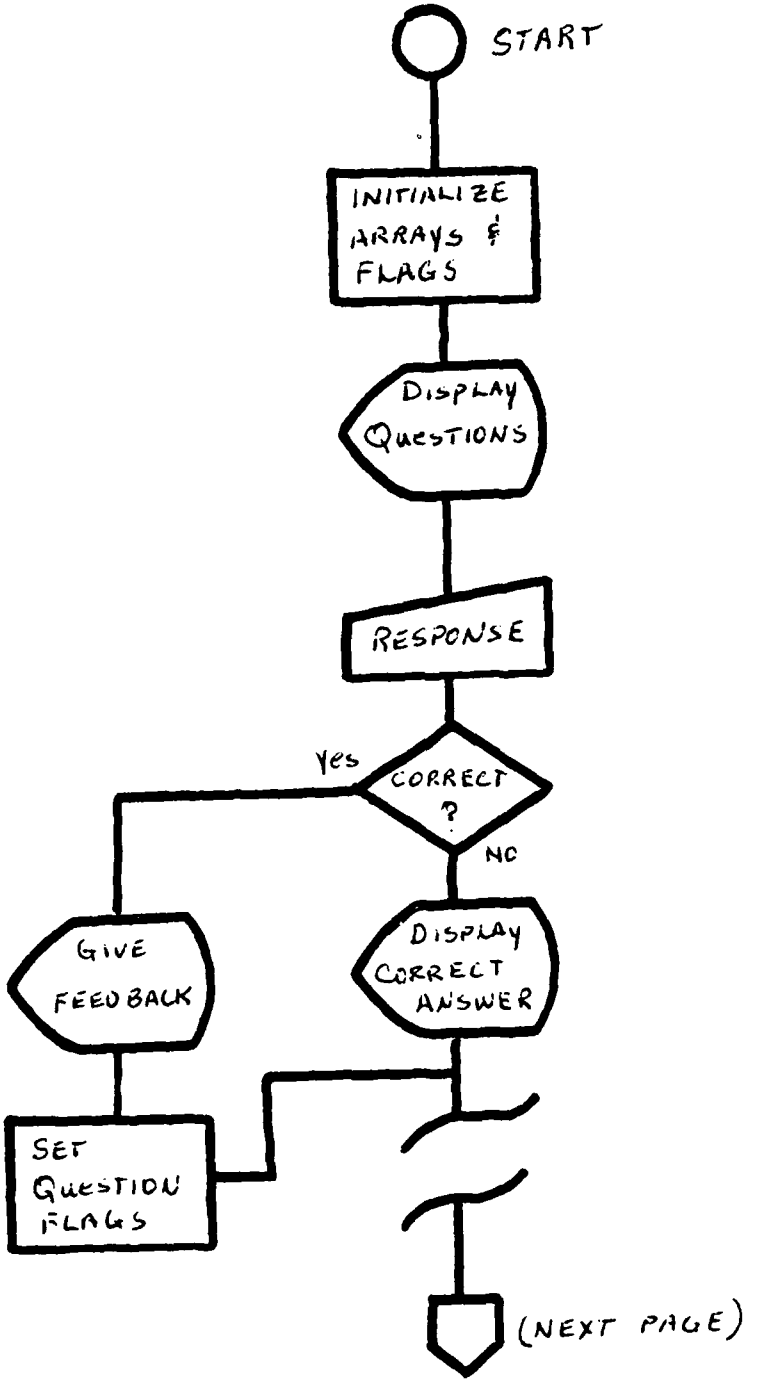


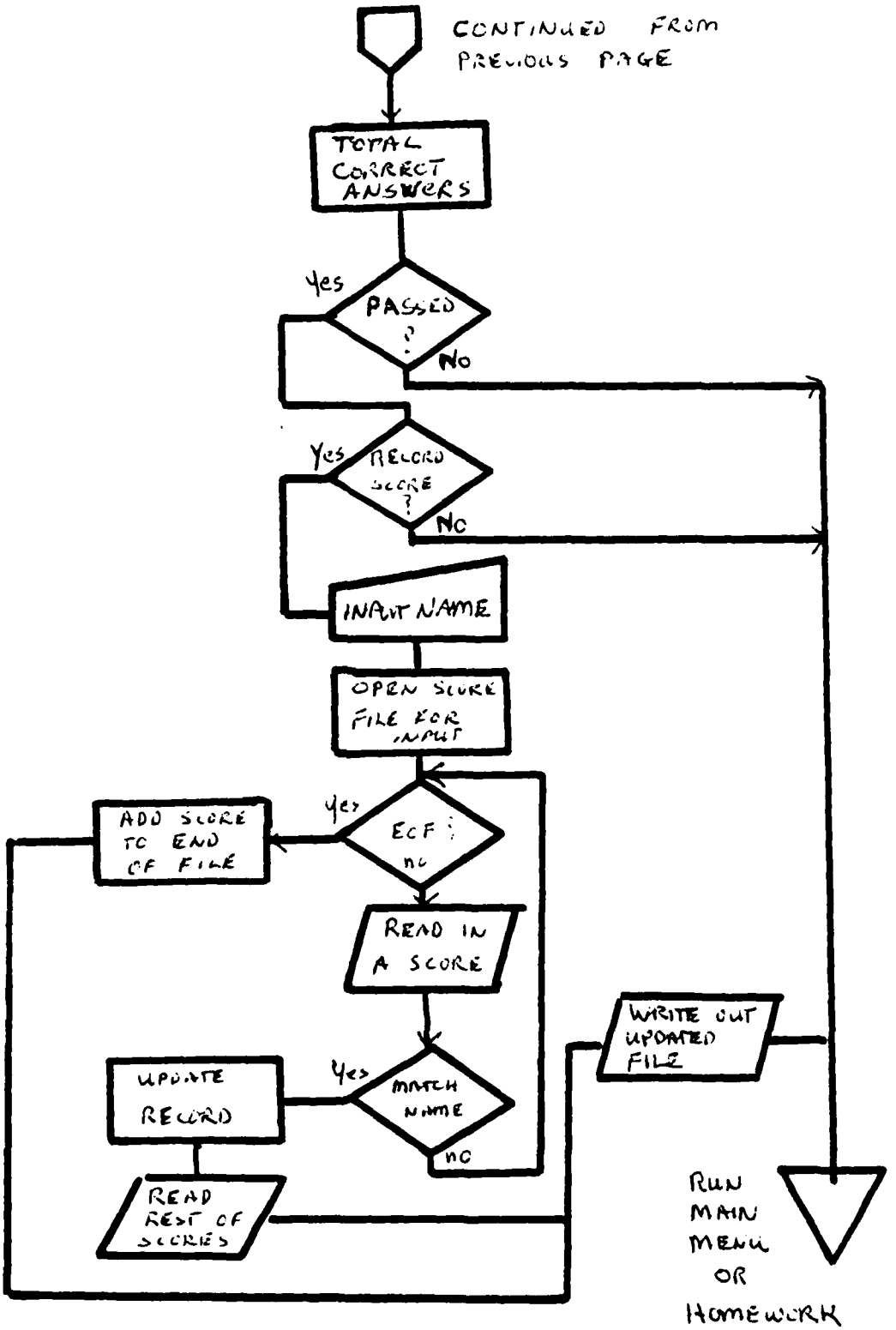
Tests 1-6

The flowcharts for the tests are all the same. Each test is 10 questions long, and seven questions must be answered correctly to pass the test. If the test is passed, the student may have the computer write their name and score to the appropriate disk file.

The score file is a sequential file, so all updates must be done by reading the entire file to memory, changing the data, and writing it back to disk. The program is dimensioned to handle 100 students.

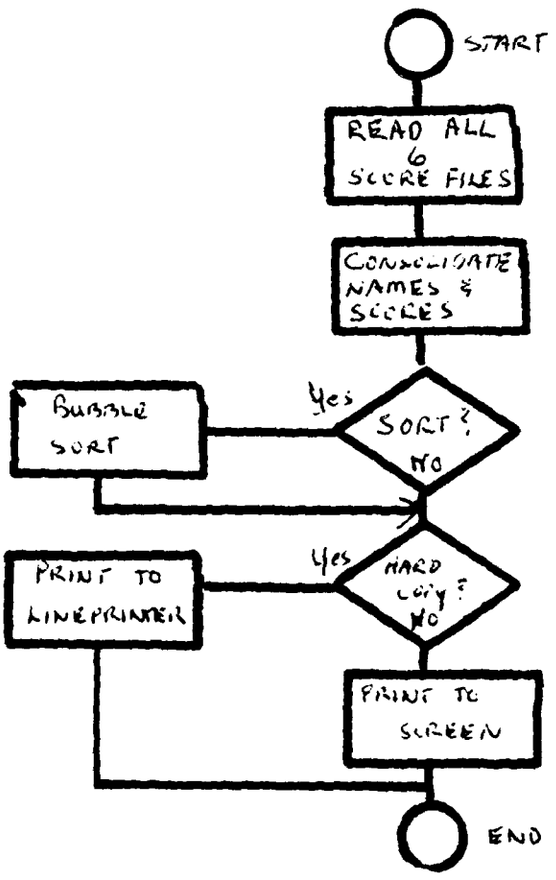
Flowchart For Tests One Through Six




CONTINUED FROM PREVIOUS PAGE
TOTAL CORRECT ANSWERS
Yes
PASSED ?
No
Yes
RECORD SCORE ?
No
INPUT NAME
OPEN SCORE FILE FOR INPUT
Yes
EOF ?
no
ADD SCORE TO END OF FILE
READ IN A SCORE
WRITE OUT UPDATED FILE
Yes
MATCH NAME
no
UPDATE RECORD
READ REST OF SCORES
RUN MAIN MENU OR HOMEWORK

Report Program

The report program reads in all the scores, assigns them to the proper students, and prints out a summary to the monitor or to the line printer. If a student hasn't completed a test for one of the lessons, a set of asterisks appears for the test score. The user of this program has the option of receiving the report with student names in alphabetical order. A sample report is on the following page.



CAI IN BASIC REPORT DATE:
07/31/83

NAME	TEST 1	TEST 2	TEST 3	TEST 4	TEST 5	TEST 6
CROMEMCO	10	9	2	9	3	***
CURLY	9	2	9	5	2	8
D.D.	4	4	6	***	***	2
DAN	***	7	5	***	5	4
DAN C.	10	7	9	***	3	7
DAVID	***	***	3	5	6	10
GOLIATH	***	5	3	6	4	5
LARRY	1	6	7	2	8	***
MOE	8	3	3	***	4	4
POSEY	***	***	3	***	***	***
TEST	5	***	***	***	***	***
WIDGET	***	6	***	***	***	***

END OF PROGRAM - HIT BREAK OR CONTROL C TO QUIT

Index of Subjects

The following index lists the main subjects, as shown in the lesson menus. The reference on the right indicates what lesson, and what part of the lesson, the subject is taught.

Program Limitations

The purpose of the CAI program is to give a student a solid core of knowledge about BASIC. From this core, the student can continue his/her own education, concentrating in any area of interest. The following areas are not fully explained in the CAI. This list could serve as an outline for further study for the student.

AREA NOT INCLUDED IN CAI	APPLICABLE BASIC WORDS
1. Formatted printing	PRINT USING
2. Direct/Rnd Access	FIELD, GET, PUT, LOF OPEN"R",1,"1fn" LSET, RSET
3. Graphics	POINT, SET, RESET (TRS-80)
4. ASCII code	SAVE "1fn",A
5. Tape input/output	CSAVE, CLOAD (TRS-80)
6. Error trapping	ON ERROR GOTO,ON ERROR GOSUB, RESUME,ERR,ERL
7. Defining precision	DEFDBL, DEFSNG, DEFINT, CVI, CVS CVD, CDBL, CSNG, CINT
8. Merging programs	MERGE
9. Screen clearing	CLS

10. Line renumbering ..... NAME (TRS-80), RENUM (C/PM)

11. Memory Modification ..... PEEK, POKE

12. Internal Clock ..... TIME$, CLOCK (TRS-80)

13. Port addressing ..... INP, OUT

14. Logical Operators ..... AND, OR, NOT

15. Variable Location in core memory ..... VARPTR

This list is not exhaustive, but indicates those areas I feel are important enough to warrent further study, but not appropriate for a beginning course in BASIC. The next chapter, Recommended Areas for Further Research, outlines some other ideas which would include some of these topics.

CHAPTER 4

RECOMMENDED AREAS FOR FURTHER RESEARCH

As we saw in chapter three, the CAI in BASIC does not transform a non-programmer into a professional. It acts as an introduction to BASIC and leaves much of the more complicated programming methods to be studied on the student's own time. It also does not teach the student about the computer operating system, special purpose programs (like word processors), about the different utilities available on a microcomputer, or about the differences between Microsoft BASIC and other dialects. These areas would be ideal for further research and development. The specific research questions would include the following:

1. Could a CAI program be mated to an audio visual presentation to increase the feedback amd student interaction; thus, increasing the potential for learning?
2. Can a CAI program be developed to teach the C/PM wordprocessing system, Wordstar?
3. Can a CAI program be developed to teach the data base manager program (dBASE II) purchased

for the Cromemco computers?

4. Can a CAI program be developed to teach the Cromemco operating system (CROMIX)?
5. Can a CAI program be developed that would teach an intermediate level of BASIC, so that programming data bases, statistical programs, and other applications can be made easier for the new programmer?
6. How can the effectiveness of the CAI in BASIC be measured once the program is in the field?
7. What are the main differences in the BASIC dialects, and can a translator program be developed that would transform a BASIC from one dialect to another? This would eliminate having to teach several different dialects to new programmers. New programs could be transformed to one dialect, say Microsoft, and then adapted for use by programmers who were proficient in that language.

Each of these areas would offer a significant improvement to keeping new people trained on the microcomputers that have become a part of the Air Force.

APPENDICES

APPENDIX A

INSTALLATION INSTRUCTIONS FOR THE CAI PROGRAM LIBRARY ON THE TRS-80 AND CROMEMCO MICROCOMPUTERS

TRS-80 INSTRUCTIONS

These instructions assume that you are familiar with the computer operating system. If you are not, ask a programmer for help in starting the lessons.

The CAI comes on three disks. Make a backup of all three disks before you attempt to use them!

This version of the CAI is built to run only on the Model IV TRS-80. However, if you have a Model II, 12, or 16, you may run it without modification if you use a modem and download the library from the three 5 inch floppy disks, or from the Cromemco double sided disks. If you want a version of the CAI that runs on the TRS-80 Model I or III, please contact me at the 416 Bombardment Wing, Aircraft Maintenance Area, Griffiss AFB, NY, after October, 1993, and I will supply you with one.

To run the program, turn on the TRS-80, put TRSDOS system disk #1 in drive 0 and press the reset button. The program will start automatically. Follow the instructions carefully.

All the programs needed for lessons one, two, three, and four are on disks one and two. Programs for lessons

five and six are on disk three. When you get to lessons two through five,you will have the choice of printing out some sample homework problems to the lineprinter. If you don't want to do that, there are examples of the homework programs in Appendix D.

The report program is on Disk #1. It uses the score files generated by all the tests to create a training report. The test scores are also on disk #1. To start the program, get to BASIC and type in RUN"REPORT". The program will lead you from there.

CROMEMCO INSTRUCTIONS

These instruction assume you are familiar with the Cromemco operating system. If you are not, ask a programmer to help you transfer the lessons to your directory.

The Cromemco lessons come on one double sided floppy disk using the CDOS operating system. Make a backup of the disk before you attempt to use it! If you are using the CROMIX multi-user system, you must use the CDOSCOPY command to copy the disk into a user directory (one way is : cdoscopy -r sfda *.bas score1 score2 score3 score4 score5 score6). If you are using CDOS, then the disk may be treated as an ordinary data disk.

If your BASIC is set up with default extensions for BASIC (ie, if you save a file from BASIC and the extension .BAS is automatically appended) the program will run as is. If you do not have automatic extensions in BASIC, then you must rename all the files so that the ".BAS" extension is removed.

To start the lessons, get the programs mounted in your system, go to BASIC, and type in RUN "MENU". The student can select a lesson and take it from there. The program

will run without anymore help.

When the homework programs are run, (lessons two through five), the student will have the option of printing out the homework assignment to the lineprinter. Be sure the lineprinter, or letter printer, is linked to the student's terminal, turned on, and loaded with at least three sheets of paper before these lessons are run.

To run the training report program, go to BASIC and type in RUN "REPORT" and follow the directions.

APPENDIX B

PROGRAM LISTINGS FOR THE CAI
PROGRAM LIBRARY

TABLE OF CONTENTS

```
10 REM ** THIS PROGRAM STARTED ON 27 MARCH 1983
15 REM ** MODIFIED FOR LDOS DOUBLE SIDED DRIVE ON 20 JUNE 1983
20 REM ** AUTHOR - CAPT D. CREAGAN - AFIT
30 REM ** TITLE - MENU PROGRAM
40 REM **
50 REM ** SUBROUTINE(S):
60 REM **
70 REM **          GOSUB 21000 = ROUTINE TO CLEAR SCREEN
80 REM **                        THE CLEAR SCREEN FUNCTION
90 REM **                        IS NON-STANDARD.  THIS
100 REM **                        SUBROUTINE PRINTS 24 LINE
110 REM **                        FEEDS TO INSURE SCREEN IS
120 REM **                        CLEARED ON MOST TERMINALS
130 REM **
140 GOSUB 520
150 PRINT"          COMPUTER ASSISTED INSTRUCTION IN BASIC"
160 PRINT
170 PRINT"             by: Captain Dan Creagan"
180 PRINT"             Air Force Institute of Technology"
190 PRINT
200 PRINT"This is the menu for computer assisted instruction"
210 PRINT"in BASIC.  It is meant to be used with a BASIC manual or with"
220 PRINT"an experienced programmer available for consultation."
230 PRINT
240 PRINT
250 PRINT"If you wish to give me feedback, or get information about"
260 PRINT"this program, please contact me at GRIFFISS AFB, NY.  I will be"
270 PRINT"in the Aircraft Maintenance area after OCT 83."
280 PRINT
290 INPUT"press ENTER to continue";T$
300 GOSUB 520
310 PRINT"                          MENU CHOICES"
320 PRINT
330 PRINT"Select the program you wish to run from the list below"
340 PRINT"and press the number that is next to your selection."
350 PRINT"Then press ENTER."
360 PRINT
370 PRINT"1.  LESSON ONE                7.  LESSON FOUR"
380 PRINT"2.  TEST ONE                  8.  TEST FOUR"
390 PRINT"3.  LESSON TWO                9.  LESSON FIVE"
400 PRINT"4.  TEST TWO                  10.  TEST FIVE"
410 PRINT"5.  LESSON THREE              11.  LESSON SIX"
420 PRINT"6.  TEST THREE                12.  TEST SIX"
425 PRINT
430 INPUT"WHICH NUMBER DO YOU WANT";T
440 ON T GOTO 450,452,454,456,460,462,464,466,470,472,474,480,490
```

```
450 RUN"LESSON1"
452 RUN"TEST1"
454 RUN"LESSON2"
456 RUN"TEST2"
460 RUN"LESSON3"
462 RUN"TEST3"
464 RUN"LESSON4"
466 RUN"TEST4"
470 RUN"LESSON5"
472 RUN"TEST5"
474 RUN"LESSON6"
480 RUN "TEST6"
490 END
500 REM **
510 REM ** FOLLOWING SUBROUTINE IS USED TO CLEAR SCREEN
520 FOR X = 1 TO 24
530     PRINT" "
540 NEXT X
550 RETURN
```

***** Listing of Program 'LESSON1' ***** 07/10/83 - 01:03:41

```
1000 REM ** THIS PROGRAM STARTED ON 27 MARCH 1983
1010 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
1020 REM ** TITLE: LESSON 1A
1030 REM **
1040 REM **
1050 REM **
1060 GOSUB 9970
1070 PRINT"LESSON:  BASIC 1A            VERSION: 1 AUGUST 83
1080 PRINT
1090 PRINT"TIME REQUIRED TO COMPLETE LESSON:  About One Hour"
1100 PRINT
1110 PRINT
1120 PRINT"AUTHOR:  Capt Danny J. Creagan"
1130 PRINT"         Air Force Institute of Technology"
1140 PRINT
1150 PRINT"OBJECTIVE:  To introduce the student to Microsoft"
1160 PRINT"            BASIC and the fundamentals of a small computer"
1170 PRINT
1180 PRINT"MATERIALS REQ'D:  BASIC reference manual"
1190 PRINT
1200 PRINT
1210 PRINT
1220 INPUT"press the ENTER key to continue";T$
1230 GOSUB 9970
1240 GOSUB 10040
1250 PRINT"A  I'm taking this part in its entirety."
1260 PRINT"B  I wish to review selected areas."
1270 PRINT"C  I want to go to the second part."
1280 PRINT"D  I want to return to the Menu."
1290 PRINT
1300 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
1310 IF T$ = "D" GOTO 10160
1320 IF T$ = "C" GOTO 10170
1330 IF T$ = "B" GOTO 1450
1340 IF T$ <>"A" GOTO 1250
1350 GOSUB 1560
1360 GOSUB 1920
1370 GOSUB 2790
1380 GOSUB 3750
1390 GOSUB 4800
1400 GOSUB 5950
1410 GOSUB 7820
1420 GOSUB 8500
1430 PRINT"GOING TO SECOND PART - PLEASE STANDBY"
1440 GOTO 10170
1450 GOSUB 9970
```

```
1460 GOSUB 10040
1470 PRINT
1480 PRINT"Please type in the number beside the area you wish"
1490 PRINT"to review (1 through 8) and then press ENTER - press 0 and"
1500 PRINT"press ENTER to return to the Menu."
1510 PRINT
1520 INPUT"What is your choice";N
1530 IF N = 0 GOTO 10160
1540 ON N GOSUB 1560 ,1920 ,2790 ,3750 ,4800 ,5950 ,7820 ,8500
1550 GOTO 1450
1560 GOSUB 9970
1570 PRINT"                    Introduction"
1580 PRINT
1590 PRINT"Throughout all your lessons, you should have your BASIC manual"
1600 PRINT"handy.  If you find yourself stumped by a question, you should"
1610 PRINT"LOOK UP THE ANSWER IN THE BOOK.  If you can't find it after"
1620 PRINT"an honest attempt, then make a guess and then go on.  You will"
1630 PRINT"have an opportunity to review each section again."
1640 PRINT
1650 PRINT"Beginning with this lesson, you will have homework assigned"
1660 PRINT"at the end of each test.  If you do the homework, you will"
1670 PRINT"learn more, and, with the techniques you learn, you will find"
1680 PRINT"that you can tackle small programming jobs as soon as you"
1690 PRINT"complete the course. "
1700 PRINT
1710 INPUT"press ENTER";T$:GOSUB 9970
1720 PRINT
1730 PRINT"                    Introduction"
1740 PRINT
1750 PRINT"Throughout the next six lessons you will be learning"
1760 PRINT"about computers and what they do.  Although the course is"
1770 PRINT"titled 'Computer Assisted Instruction in BASIC', you will"
1780 PRINT"also need to learn the terminology of computers, not just"
1790 PRINT"the BASIC programming language.  This first lesson will"
1800 PRINT"start with some fundamental ideas, and expand them as"
1810 PRINT"we go along."
1820 PRINT
1830 PRINT"We use computers to process DATA and give us answers to our"
1840 PRINT"problems. To process this DATA, we must communicate with the "
1850 PRINT"computer using two basic computer components.  Those com-"
1860 PRINT"are called:       HARDWARE AND SOFTWARE."
1870 PRINT
1880 INPUT"press ENTER";T$
1890 GOSUB 10210
1900 IF T$ = "B" GOTO 1560
1910 RETURN
```

```
1920 GOSUB 9970
1930 PRINT"                        HARDWARE"
1940 PRINT
1950 PRINT"Hardware is the term used to describe the electrical and"
1960 PRINT"mechanical aspects of a computer.  Hardware includes the"
1970 PRINT"parts you can physically touch on, or in, your computer."
1980 PRINT
1990 PRINT"One major piece of hardware is the central processing unit"
2000 PRINT"(CPU). The CPU is the computer's central electronic brain."
2010 PRINT
2020 PRINT"It performs all of the data operations and contains a"
2030 PRINT"storage area called MEMORY which is used for short term"
2040 PRINT"data retention during operations."
2050 PRINT
2060 INPUT"press ENTER to continue";T$
2070 GOSUB 9970
2080 PRINT"                    HARDWARE (cont)
2090 PRINT
2100 PRINT"PERIPHERAL DEVICES are additional units of equipment"
2110 PRINT"that support the computer. PERIPHERAL DEVICES are used for"
2120 PRINT"long-term or permanent storage, and they also let"
2130 PRINT"you communicate with the computer."
2140 PRINT
2150 PRINT"The computer 'talks' to you by using peripheral"
2160 PRINT"hardware units called OUTPUT devices.  These can be"
2170 PRINT"TERMINALS, or LINE PRINTERS, or TAPES, or DISKS."
2180 PRINT
2190 PRINT"You 'talk' to the computer through units called INPUT devices."
2200 PRINT
2210 INPUT"press ENTER to continue";T$
2220 GOSUB 9970
2230 PRINT"                HARDWARE (cont)
2240 PRINT
2250 PRINT"INPUT DEVICES may also be terminals, or tape, or disks,"
2260 PRINT"or, in some special cases, printers that have keyboards"
2270 PRINT"that are used as terminals."
2280 PRINT
2290 PRINT"INPUT and/or OUTPUT DEVICES provide a physical communication"
2300 PRINT"link between you and the computer."
2310 PRINT
2320 PRINT"Whenever you communicate, there must be something that"
2330 PRINT"transforms your physical requests (key strokes) into "
2340 PRINT"electrical DATA that the CPU understands. Most of that"
2350 PRINT"job is done by SOFTWARE"
2360 PRINT
2370 INPUT"press ENTER to continue";T$
```

```
2380 GOSUB 9970
2390 PRINT"Here is a little quiz - answer in capital letters"
2400 PRINT"and do not include extra spaces or words"
2410 PRINT
2420 PRINT"What component (HARDWARE or SOFTWARE) is mainly"
2430 PRINT"used to transform your inputs into a form the CPU can"
2440 INPUT"understand";T$
2450 PRINT
2460 IF T$ <> "SOFTWARE" THEN PRINT "WRONG - the correct answer is SOFTWARE"
2470 IF T$ ="SOFTWARE" THEN PRINT"CORRECT"
2480 PRINT
2490 INPUT"press ENTER to continue";T$
2500 GOSUB 9970
2510 INPUT"What do the letters CPU stand for";T$
2520 PRINT
2530 IF T$ = "CENTRAL PROCESSING UNIT" GOTO 2560
2540 PRINT"WRONG - the correct answer is CENTRAL PROCESSING UNIT"
2550 GOTO 2570
2560 PRINT"CORRECT - now we are learning something!"
2570 PRINT
2580 INPUT"press ENTER to continue";T$
2590 GOSUB 9970
2600 PRINT"Which of the following can be considered an OUTPUT device?"
2610 PRINT
2620 PRINT"A  Terminals"
2630 PRINT"B  Tapes"
2640 PRINT"C  Disks"
2650 PRINT"D  ALL of the above"
2660 PRINT
2670 INPUT"which letter do you select";T$
2680 IF LEN(T$)<>1 GOTO 2670
2690 PRINT
2700 IF T$ = "D" GOTO 2730
2710 PRINT"WRONG - D (ALL of the above) is the correct answer"
2720 GOTO 2740
2730 PRINT"CORRECT"
2740 PRINT
2750 INPUT"press ENTER to continue";T$
2760 GOSUB 10210
2770 IF T$ = "B" GOTO 1920
2780 RETURN
2790 GOSUB 9970
2800 PRINT"                    SOFTWARE"
2810 PRINT
2820 PRINT"Software is a collection of written rules that control"
2830 PRINT"the computer.  Software can be divided into two"
```

```
2840 PRINT"types:  USER PROGRAMS and OPERATING SYSTEMs."
2850 PRINT
2860 PRINT"A USER PROGRAM is the instructions that you write to"
2870 PRINT"the machine that tell it where your data is, what to do with"
2880 PRINT"it, and when to do it."
2890 PRINT
2900 PRINT"The OPERATING SYSTEM is the software that is the conscious-"
2910 PRINT"ness of the computer."
2920 PRINT
2930 INPUT"press ENTER to continue";T$
2940 GOSUB 9970
2950 PRINT"                SOFTWARE (cont)"
2960 PRINT
2970 PRINT"The OPERATING SYSTEM supervises the various capabilities "
2980 PRINT"of the computer and cannot be altered by the user."
2990 PRINT"It OVERSEES the operation, and senses when a keystroke"
3000 PRINT"is made, a button is pushed, or a request made."
3010 PRINT
3020 PRINT"One part of the operating system is called the LANGUAGE"
3030 PRINT"PROCESSOR. The LANGUAGE PROCESSOR translates the instructions"
3040 PRINT"of a user-written program into electronic instruc-"
3050 PRINT"tions that the computer can understand."
3060 PRINT
3070 PRINT"The rules, or grammar, that you use to write your software"
3080 PRINT"are described by the kind of computer language you use."
3090 PRINT
3100 INPUT"press ENTER to continue";T$
3110 GOSUB 9970
3120 PRINT"                  SOFTWARE (cont)"
3130 PRINT
3140 PRINT"Because user-written programs and operating systems are"
3150 PRINT"both designed by humans, it is possible to develop a human"
3160 PRINT"oriented language that both can use.  That is, a language "
3170 PRINT"that lets you write programs using easily mastered rules and"
3180 PRINT"conventions that are also understood by the operating"
3190 PRINT"system. Once we get the operating system to understand"
3200 PRINT"the instruction, it can make the computer do its job."
3210 PRINT
3220 PRINT
3230 PRINT
3240 PRINT
3250 PRINT
3260 INPUT"press ENTER to continue";T$
3270 GOSUB 9970
3280 PRINT"It's time for another quiz!"
3290 PRINT
```

```
3300 PRINT"Remember, use only capital letters and don't add unnecessary"
3310 PRINT"spaces or words."
3320 PRINT
3330 INPUT"press ENTER to continue";T$
3340 GOSUB 9970
3350 PRINT"Is a peripheral, such as a line printer, hardware or software?"
3360 PRINT
3370 PRINT"A  Hardware"
3380 PRINT"B  Software"
3390 PRINT
3400 PRINT"Choose A or B - press the letter and then press ENTER"
3410 PRINT
3420 INPUT"Which letter";T$
3430 IF LEN(T$) > 1 GOTO 3400
3440 PRINT
3450 IF T$ = "A" GOTO 3480
3460 PRINT"WRONG - the correct answer is A (Hardware)."
3470 GOTO 3490
3480 PRINT"CORRECT"
3490 PRINT
3500 INPUT"press ENTER to continue";T$
3510 GOSUB 9970
3520 PRINT"An operating system translates user-written code into a"
3530 PRINT"form that the operator can understand. (TRUE or"
3540 PRINT"FALSE?)"
3550 PRINT
3560 PRINT"A  True"
3570 PRINT"B  False"
3580 PRINT
3590 PRINT"Choose the letter corresponding to the correct answer"
3600 PRINT
3610 INPUT"which letter (A or B)";T$
3620 PRINT
3630 IF T$ = "B" GOTO 3690
3640 PRINT"INCORRECT - the right answer is B (False)"
3650 PRINT
3660 PRINT"It is the language processor's job to convert user code"
3670 PRINT"into a form the computer can understand!"
3680 GOTO 3700
3690 PRINT"You are RIGHT"
3700 PRINT
3710 INPUT"press ENTER to continue";T$
3720 GOSUB 10210
3730 IF T$ = "B" GOTO 2790
3740 RETURN
3750 GOSUB 997
```

```
3750 PRINT"                  General Information"
3770 PRINT
3780 PRINT"There have been many programming languages developed over the"
3790 PRINT"years. Many were designed to solve specific problems and they"
3800 PRINT"required a good deal of previous knowledge about computers."
3810 PRINT
3820 PRINT
3830 PRINT"BASIC, which stands for Beginner's All-purpose Symbolic "
3840 PRINT"Instruction Code, is a language that requires only a"
3850 PRINT"moderate understanding of how a computer works."
3860 PRINT
3870 PRINT"BASIC was developed at Dartmouth College for use by students"
3880 PRINT"who were unfamiliar with computers and needed a language"
3890 PRINT"related to everyday speech."
3900 PRINT
3910 INPUT"press ENTER to continue";T$
3920 GOSUB 9970
3930 PRINT"               General Information (cont)"
3940 PRINT
3950 PRINT"BASIC is easier to master than most other languages, be-"
3960 PRINT"cause its instructions are very similar to  English grammar."
3970 PRINT
3980 PRINT"However, BASIC is not English.  A computer must be instruc-"
3990 PRINT"ed in precise terms, with no ambiguity.  English has many"
4000 PRINT"synonymous and imprecise terms."
4010 PRINT
4020 PRINT
4030 PRINT
4040 PRINT
4050 INPUT"press ENTER for more";T$
4060 GOSUB 9970
4070 PRINT"           General Information (cont)"
4080 PRINT
4090 PRINT"To further explain the difference between BASIC and"
4100 PRINT"English, if you describe how to average numbers in English"
4110 PRINT"you might do it this way. (assuming the numbers below)"
4120 PRINT
4130 PRINT"   Add 19, 80, 50 100, and 56.  Divide by 5."
4140 PRINT"   Write the quotient as the answer."
4150 PRINT
4160 PRINT"A computer programmed in BASIC couldn't understand these"
4170 PRINT"instructions; however, the instructions that BASIC would"
4180 PRINT"use are very similar to these.  BASIC just distills down"
4190 PRINT"the commands and eliminates all the ambiguity."
4200 PRINT"This average can be stated in one instruction called PRINT."
4210 PRINT
```

```
4220 INPUT"press ENTER for an example of the PRINT instruction";T$
4230 GOSUB 9970
4240 PRINT"              General Information (cont)"
4250 PRINT
4260 PRINT"The PRINT statement works like this, to find the average"
4270 PRINT"of five numbers and write the result on your terminal."
4280 PRINT"you can use the following BASIC statement:"
4290 PRINT
4300 PRINT"          PRINT (19+80+50+100+60)/5"
4310 PRINT
4320 PRINT"In this example, the BASIC verb PRINT tells the operating"
4330 PRINT"system to write the instruction following it to the terminal."
4340 PRINT"The data, or recipients of the verb PRINT, are the numbers"
4350 PRINT"and symbols to the right of the PRINT word. The symbols"
4360 PRINT"are used the same way that you use them on a calculator."
4370 PRINT
4380 PRINT
4390 INPUT"press ENTER to continue with General Information";T$
4400 GOSUB 9970
4410 PRINT"BASIC is used by nearly every micro and mainframe computer."
4420 PRINT
4430 PRINT"There are many 'dialects' of BASIC and they are not all"
4440 PRINT"compatible with each other.  For instance, a BASIC program"
4450 PRINT"written in Honeywell BASIC or Applesoft BASIC will not"
4460 PRINT"run on your computer unless it is modified."
4470 PRINT
4480 PRINT"The name for the BASIC on your computer is Microsoft BASIC."
4490 PRINT"Microsoft BASIC is supported by more microcomputers than"
4500 PRINT"any other dialect.  It is extremely powerful, and matches"
4510 PRINT"the computing capability of most other languages. It"
4520 PRINT"is slower than some, but the slowness is relative (most"
4530 PRINT"calculations only take milliseconds)."
4540 PRINT
4550 INPUT"press ENTER";T$
4560 GOSUB 9970
4570 PRINT
4580 PRINT"Why is BASIC a good general programming language to learn?"
4590 PRINT
4600 PRINT"A  It can be used by most students and programmers"
4610 PRINT"B  You don't need to know a lot about computers to use it"
4620 PRINT"C  It is available on most computers"
4630 PRINT"D  ALL of the above"
4640 PRINT
4650 PRINT"Press the letter corresponding to the correct answer"
4660 PRINT"Be sure to enter only capital letters"
4670 PRINT
```

```
4630 INPUT"What is your selection";T$
4690 PRINT
4700 IF T$ = "D" GOTO 4740
4710 PRINT"WRONG - the correct answer is D 'ALL of the above)"
4720 PRINT
4730 GOTO 4750
4740 PRINT"CORRECT"
4750 PRINT
4760 INPUT"press ENTER to continue";T$
4770 GOSUB 10210
4780 IF T$ = "9" GOTO 3750
4790 RETURN
4800 GOSUB 9970
4810 PRINT"                STATEMENTS and PROGRAMS"
4820 PRINT
4830 PRINT"The instruction that we saw in the previous example is a"
4840 PRINT"one line command to the computer. When we combine several"
4850 PRINT"statements, we get a more useful COMPUTER PROGRAM."
4860 PRINT
4870 PRINT"The COMPUTER PROGRAM acts as a series of directions for"
4880 PRINT"the machine to follow."
4890 PRINT
4900 PRINT"The statements that make up the program are expressed as"
4910 PRINT"BASIC verbs which denote an action to be taken.  THEY"
4920 PRINT"APPEAR SEQUENTIALLY ON NUMBERED PROGRAM LINES, usually"
4930 PRINT"along with the data that is to be acted upon."
4940 PRINT
4950 INPUT"press ENTER to continue with STATEMENTS & PROGRAMS";T$
4960 GOSUB 9970
4970 PRINT
4980 PRINT"Each BASIC statement consists of a specific arrangement of"
4990 PRINT"elements.  These elements are shown below, in the order"
5000 PRINT"they MUST appear in an actual program line"
5010 PRINT
5020 PRINT"     STATEMENT (or LINE) NUMBER"
5030 PRINT"         - indicates the processing sequence of the statements"
5040 PRINT"         - always in ascending order."
5050 PRINT"     BASIC WORD"
5060 PRINT"         - specifies the computer operation to be performed"
5070 PRINT"     PARAMETERS"
5080 PRINT"         - variables, or expressions,"
5090 PRINT"           used to direct the operation performed"
5100 PRINT"           by the statement."
5110 PRINT
5120 INPUT" press ENTER to continue";T$
5130 GOSUB 9970
```

```
5140 PRINT"            STATEMENTS and PROGRAMS (cont)"
5150 PRINT
5160 PRINT"Every statement must have a line number and these numbers range"
5170 PRINT"from 0 to 65529 in most microcomputers that support"
5180 PRINT"Microsoft BASIC.  Microsoft is the company that owns"
5190 PRINT"the copyright to the particular dialect of BASIC that"
5200 PRINT"runs on this machine).  It is advisable to write program"
5210 PRINT"lines in increments of 10 to allow you to insert additional"
5220 PRINT"lines without having to renumber every statement line."
5230 PRINT"The statements are executed in ascending numerical order."
5240 PRINT"not in the order they were entered."
5250 PRINT
5260 PRINT"Look up the RENUM, or NAME command in your manual for more"
5270 PRINT"information!"
5280 PRINT
5290 INPUT"press ENTER to continue";T$
5300 GOSUB 9970
5310 PRINT"              STATEMENTS and PROGRAMS (cont)"
5320 PRINT
5330 PRINT"The last statement of the program should be the END statement."
5340 PRINT"This indicates that the program is complete.  IT IS NOT "
5350 PRINT"NECESSARY, but it is a good practice to always put it in."
5360 PRINT"
5370 PRINT"To get the program to execute you use the RUN command."
5380 PRINT
5390 PRINT"Now for a few questions to see how you are doing."
5400 PRINT
5410 INPUT"press ENTER for the questions";T$
5420 GOSUB 9970
5430 PRINT"A computer program is a series of ---------"
5440 PRINT
5450 PRINT"A  Verbs"
5460 PRINT"B  Words"
5470 PRINT"C  Statements"
5480 PRINT"D  Synonyms"
5490 PRINT
5500 PRINT
5510 INPUT"Type in the letter opposite the correct answer and press ENTER";T$
5520 PRINT
5530 IF T$ = "C" GOTO 5560
5540 PRINT"WRONG - the correct answer is C (STATEMENTS)."
5550 GOTO 5570
5560 PRINT"CORRECT"
5570 PRINT
5580 INPUT"press ENTER to continue";T$
5590 GOSUB 9970
```

```
5600 PRINT"Which of the following is incorrect?"
5610 PRINT
5620 PRINT"A  A BASIC word is a word that a BASIC processor understands"
5630 PRINT"B  A statement can have no more than two line numbers"
5640 PRINT"C  Data are the recipients of the action of BASIC verbs"
5650 PRINT"D  Line Numbers are written sequentially."
5660 PRINT
5670 PRINT"Press the letter that is beside the correct answer"
5680 PRINT"and then press ENTER"
5690 PRINT
5700 INPUT"What is your choice";T$
5710 PRINT
5720 IF T$ = "B" GOTO 5750
5730 PRINT"WRONG - B Only one line number is allowed per statement"
5740 GOTO 5760
5750 PRINT"CORRECT - way to go!"
5760 PRINT
5770 INPUT"press ENTER for the next question";T$
5780 GOSUB 9970
5790 PRINT"Is 350000 a valid statement number in Microsoft BASIC?"
5800 PRINT
5810 PRINT"N  No"
5820 PRINT"Y  Yes"
5830 PRINT
5840 INPUT"press the letter beside the correct answer and then press ENTER";T$
5850 PRINT
5860 IF T$ = "N" GOTO 5890
5870 PRINT"WRONG - 350000 is too big. Remember, you can only go to 65529"
5880 GOTO 5900
5890 PRINT"CORRECT"
5900 PRINT
5910 INPUT"press ENTER";T$
5920 GOSUB 10210
5930 IF T$ = "S" GOTO 4800
5940 RETURN
5950 GOSUB 9970
5960 PRINT"                    PRINT STATEMENT"
5970 PRINT
5980 PRINT"The BASIC word PRINT is a command that tells the computer"
5990 PRINT"to output the data that follows to the computer terminal"
6000 PRINT
6010 PRINT"This data can be numbers, variables, or strings."
6020 PRINT"Strings are combinations of words or numbers that are"
6030 PRINT"to be printed without having any calculations done to them"
6040 PRINT
6050 PRINT
```

```
6060 PRINT
6070 PRINT
6080 INPUT"press ENTER";T$
6090 GOSUB 9970
6100 PRINT"You can control the output caused by the print statement"
6110 PRINT"in two ways.  If you just want what you ENTER printed"
6120 PRINT"without any calculations done to it, then you enclose the"
6130 PRINT"data after the PRINT command in quotation marks."
6140 PRINT
6150 PRINT"For Example:"
6160 PRINT
6170 PRINT"10 PRINT "CHR$(34)"Go For It!"CHR$(34)
6180 PRINT"20 END
6190 PRINT"RUN"
6200 PRINT
6210 PRINT"In BASIC, if you type this in as shown, you get this result:"
6220 PRINT
6230 PRINT"Go For It!"
6240 PRINT
6250 INPUT"press ENTER";T$
6260 GOSUB 9970
6270 PRINT"                         PRINT (cont)"
6280 PRINT
6290 PRINT"Another example would be:"
6300 PRINT
6310 PRINT"10 PRINT "CHR$(34)"This is easy"CHR$(34)
6320 PRINT"20 END"
6330 PRINT"RUN"
6340 PRINT
6350 PRINT"Which would result in:"
6360 PRINT
6370 PRINT"This is easy"
6380 PRINT
6390 PRINT"Notice that nothing is changed by the computer, the words"
6400 PRINT"that were commanded to be output were printed exactly as shown."
6410 PRINT
6420 INPUT"press ENTER for the second example";T$
6430 GOSUB 9970
6440 PRINT"                    PRINT (cont)"
6450 PRINT
6460 PRINT"The second way the print statement is used to control output"
6470 PRINT"is by NOT enclosing the data in quotation marks.  The data is"
6480 PRINT"then read by the computer and the computer tries to evaluate"
6490 PRINT"what the data means in mathematical terms.  If you have entered"
6500 PRINT"data that cannot be mathematically manipulated, then you"
6510 PRINT"get an ERROR message."
```

```
6520 FOR X = 1 TO 7
6530  PRINT
6540 NEXT X
6550 INPUT"press ENTER";T$
6560 GOSUB 9970
6570 PRINT
6580 PRINT"Here is an example of data in quotes, and data not in quotes"
6590 PRINT"and what the output would look like:"
6600 PRINT
6610 PRINT
6620 PRINT
6630 PRINT"10 PRINT "CHR$(34)"1+1"CHR$(34)
6640 PRINT"20 PRINT 1+1"
6650 PRINT"RUN"
6660 PRINT
6670 PRINT"Results in:"
6680 PRINT
6690 PRINT"1+1"
6700 PRINT"2"
6710 PRINT
6720 INPUT"press ENTER";T$
6730 GOSUB 9970
6740 PRINT"                    PRINT (cont)"
6750 PRINT
6760 PRINT"10 PRINT "CHR$(34)"1+1"CHR$(34)
6770 PRINT"20 PRINT 1+1"
6780 PRINT
6790 PRINT "1+1"
6800 PRINT "2"
6810 PRINT
6820 PRINT"Note that the statement that had quotes was reprinted exactly"
6830 PRINT"as it was typed in, without the quotes, while the second"
6840 PRINT"statement was computed mathematically and a result was given."
6850 PRINT"The part of the first statement within quotes is called"
6860 PRINT"a STRING - remember?!"
6870 PRINT
6880 INPUT"press ENTER";T$
6890 GOSUB 9970
6900 PRINT"                         PRINT (cont)"
6910 PRINT
6920 PRINT"The format of the output of PRINT can be controlled using"
6930 PRINT"commas.  For Example:"
6940 PRINT
6950 PRINT"10 PRINT "CHR$(34)"A"CHR$(34)",6+2,"CHR$(34)"B"CHR$(34)",7+1"
6960 PRINT"20 END"
6970 PRINT"RUN"
```

```
6980 PRINT
6990 PRINT"Results in:"
7000 PRINT
7010 PRINT"A",6+2,"B",7+1
7020 PRINT
7030 PRINT
7040 PRINT
7050 INPUT"press ENTER";T$
7060 GOSUB 9970
7070 PRINT"A",6+2,"B",7+1
7080 PRINT
7090 PRINT"Notice how the commas have caused the terms of the"
7100 PRINT"statement to be spaced across the screen.  The spaces"
7110 PRINT"are similar to TABS on a typewriter, however, the comma reacts"
7120 PRINT"differently on different terminals.  See your BASIC manual"
7130 PRINT"or ask your system operator how they react on your machine."
7140 PRINT
7150 PRINT"(commas usually cause 8 spaces between terms)"
7160 PRINT
7170 INPUT"press ENTER";T$
7180 GOSUB 9970
7190 PRINT"                         PRINT (cont)"
7200 PRINT
7210 PRINT"The PRINT statement also allows you to output blank lines."
7220 PRINT"You print blank lines by typing in the line number and then "
7230 PRINT"a PRINT statement without an argument.  For example:"
7240 PRINT
7250 PRINT"10 PRINT "CHR$(34)"Now is the time to skip"CHR$(34)
7260 PRINT"20 PRINT"
7270 PRINT"30 PRINT "CHR$(34)"a line."CHR$(34)
7280 PRINT"RUN"
7290 PRINT
7300 INPUT"press ENTER for results";T$
7310 PRINT
7320 PRINT"Now is the time to skip"
7330 PRINT
7340 PRINT"a line"
7350 PRINT
7360 INPUT"press ENTER";T$
7370 GOSUB 9970
7380 PRINT"12+10"
7390 PRINT
7400 PRINT"Which of the following statements would cause the above output?"
7410 PRINT
7420 PRINT"A  10 PRINT "CHR$(34)"12+10"CHR$(34)
7430 PRINT
```

```
7440 PRINT"B  20 PRINT 12+10"
7450 PRINT
7460 PRINT"C  15 PRINT "CHR$(34)"TWELVE + TEN"CHR$(34)
7470 PRINT
7480 PRINT"D  25 PRINT 12+10"
7490 PRINT
7500 INPUT"press the letter that is beside the correct choice and ENTER";T$
7510 PRINT
7520 IF T$ = "A" GOTO 7630
7530 PRINT"WRONG - the correct answer is A"
7540 PRINT
7550 PRINT"10 PRINT "CHR$(34)"12+10"CHR$(34)
7560 PRINT"RUN"
7570 PRINT
7580 PRINT "Which results in:"
7590 PRINT
7600 PRINT "12+10"
7610 PRINT
7620 GOTO 7640
7630 PRINT"CORRECT - that was a key concept, congratulations!"
7640 PRINT
7650 INPUT"press ENTER";T$
7660 GOSUB 9970
7670 PRINT"Write the statement that would cause a blank line to be printed"
7680 PRINT"Use 10 for the line number and leave one blank space between"
7690 PRINT"terms."
7700 PRINT
7710 INPUT"What is your answer";T$
7720 PRINT
7730 IF T$ = "10 PRINT" GOTO 7760
7740 PRINT"WRONG - the correct answer is:  10 PRINT"
7750 GOTO 7770
7760 PRINT "CORRECT"
7770 PRINT
7780 INPUT "press ENTER to continue";T$
7790 GOSUB 10210
7800 IF T$ = "B" GOTO 5950
7810 RETURN
7820 GOSUB 9970
7830 PRINT"                    END & STOP"
7840 PRINT
7850 PRINT "The END statement is the last statement in a program."
7860 PRINT"It notifies the computer when the program is done."
7870 PRINT "Because it is the last statement, it has the highest"
7880 PRINT"line number.  The END statement is not necessary in Microsoft"
7890 PRINT"BASIC, but many programmers use it anyway.  They believe a"
```

```
7900 PRINT"program is more understandable and easier to 'track' by"
7910 PRINT"another programmer if there is only ONE entry and ONE exit in"
7920 PRINT"a program."
7930 PRINT
7940 PRINT
7950 INPUT"press ENTER for the rest of END & STOP";T$
7960 GOSUB 9970
7970 PRINT"The STOP statement interrupts execution of the program."
7980 PRINT"It is primarily used as a debugging aid.  If you want"
7990 PRINT"to find the status of a variable at a certain point in a"
8000 PRINT"program, you insert a STOP statement. For example:"
8010 PRINT
8020 PRINT"10 X = 2+3"
8030 PRINT"20 Y = X/5"
8040 PRINT"30 STOP"
8050 PRINT"40 X = Y+2"
8060 PRINT
8070 PRINT"When this program is RUN it will STOP execution at line 30"
8080 PRINT"Then you may ask the computer to tell you the status of any"
8090 PRINT"of the variables X or Y.  You can do this using the IMMEDIATE"
8100 PRINT"mode (explained next section).  Simply type in PRINT X,Y."
8110 PRINT
8120 INPUT"press ENTER";T$
8130 GOSUB 9970
8140 PRINT"If you want to start the program back up from where you"
8150 PRINT"STOPed it, then type in CONT (CONTinue) and press ENTER"
8160 PRINT"For example:"
8170 PRINT
8180 PRINT"10 X = 12+5"
8190 PRINT"20 Y = 3+2"
8200 PRINT"30 STOP
8210 PRINT"40 PRINT X+Y"
8220 PRINT"RUN"
8230 PRINT
8240 PRINT"Results in:"
8250 PRINT
8260 PRINT"BREAK IN 30"
8270 PRINT
8280 PRINT
8290 INPUT"hit ENTER for the rest";T$
8300 GOSUB 9970
8310 PRINT"BREAK in 30"
8320 PRINT
8330 PRINT
8340 PRINT"Now, by typing in CONT, the program will resume execution"
8350 PRINT"For example:"
```

```
8360 PRINT
8370 PRINT"CONT"
8380 PRINT"22"
8390 PRINT
8400 PRINT
8410 PRINT"Notice how the last line (which was PRINT X+Y) was executed?"
8420 PRINT"It was just as if the STOP statement had never been there"
8430 PRINT"As you progress in BASIC, you will find many uses for this"
8440 PRINT"statement."
8450 PRINT
8460 INPUT"press ENTER to continue";T$
8470 GOSUB 10210
8480 IF T$ = "B" GOTO 7820
8490 RETURN
8500 GOSUB 9970
8510 PRINT"                    IMMEDIATE MODE"
8520 PRINT
8530 PRINT"Microsoft BASIC has a mode called IMMEDIATE. . . ."
8540 PRINT"Whenever you have implemented BASIC in your system, usually by"
8550 PRINT"typing in the word BASIC, (consult your BASIC manual or your"
8560 PRINT"system operator for specific directions on your particular"
8570 PRINT"machine) you will be in the IMMEDIATE mode.  In this"
8580 PRINT"mode, you may execute many BASIC statements without having to"
8590 PRINT"type in line numbers or the command RUN.  For example:"
8600 PRINT
8610 PRINT"PRINT "CHR$(34)"JUST WHEN I THOUGHT I HAD THE HANG OF IT"CHR$(34)
8620 PRINT
8630 PRINT"This line will print the statement within the quotes as soon"
8640 PRINT"as the ENTER key is pressed."
8650 PRINT
8660 INPUT"press ENTER to continue";T$
8670 GOSUB 9970
8680 PRINT"                    IMMEDIATE (cont)"
8690 PRINT
8700 PRINT"Another example would be:"
8710 PRINT
8720 PRINT"PRINT 93+10+40
8730 PRINT
8740 PRINT"Which would result in:"
8750 PRINT
8760 PRINT"143"
8770 PRINT
8780 PRINT"As you can see, the computer will do the calculations just"
8790 PRINT"as if it were commanded to do it in the normal way."
8800 PRINT
8810 PRINT
```

```
8820 PRINT
8830 INPUT"press ENTER";T$
8840 GOSUB 9970
8850 PRINT"                    IMMEDIATE (cont)"
8860 PRINT
8870 PRINT"The biggest disadvantage of the IMMEDIATE mode is that the data"
8880 PRINT "is not stored in memory, and cannot be repeated again."
8890 PRINT "It is lost after the initial display, whereas the programs we"
8900 PRINT "looked at before can be run over and over again by merely "
8910 PRINT "typing in the word RUN.  Also, the immediate mode is limited to"
8920 PRINT "one line of statements at a time."
8930 PRINT
8940 PRINT
8950 INPUT"press ENTER";T$
8960 GOSUB 9970
8970 PRINT"                    NEW Statement"
8980 PRINT
8990 PRINT"If you want to clean out the temporary memory in BASIC, all"
9000 PRINT"you have to do is type in the word NEW.  BUT BE CAREFUL WITH"
9010 PRINT"THIS COMMAND.  It will erase any program you have resident."
9020 PRINT
9030 PRINT
9040 PRINT"Let's say you've been practicing the commands you have learned"
9050 PRINT"so far, and you have put in a lot of line numbers and RUN them."
9060 PRINT"But now you want to start over.  You can erase the mess with"
9070 PRINT"the command NEW."
9080 PRINT
9090 INPUT"press ENTER";T$
9100 GOSUB 9970
9110 PRINT"                         LIST"
9120 PRINT
9130 PRINT"Suppose you don't know what's there and you want to find out?"
9140 PRINT"Just type in the command LIST.  LIST will show you everything"
9150 PRINT"that's in temporary memory."
9160 PRINT
9170 INPUT"press ENTER";T$
9180 GOSUB 9970
9190 PRINT"                       DELETE"
9200 PRINT
9210 PRINT"Finally, what if you don't want to type in a new program,"
9220 PRINT"you just want to delete a line?  You can do that by typing in"
9230 PRINT"DELETE 10, or DELETE 20 or DELETE (line number)."
9240 PRINT
9250 PRINT"If you want to delete a range of line numbers, you type in"
9260 PRINT"DELETE (low range-high range).  Say you want to delete lines"
9270 PRINT"15 to 35.  You would type in DELETE 15-35, and the lines"
```

```
9280 PRINT"would no longer be in memory!"
9290 PRINT
9300 INPUT"press ENTER";T$
9310 GOSUB 9970
9320 PRINT"Which of the following statements would be an example of"
9330 PRINT"IMMEDIATE mode in Microsoft BASIC?"
9340 PRINT
9350 PRINT"A  10 PRINT "CHR$(34)"AT LAST"CHR$(34)
9360 PRINT"   RUN"
9370 PRINT
9380 PRINT"B  PRINT 25+2-3"
9390 PRINT
9400 PRINT"C  32+2"
9410 PRINT
9420 PRINT"D  STOP"
9430 PRINT
9440 INPUT"press the letter beside the correct answer and press ENTER";T$
9450 PRINT
9460 IF T$ = "B" GOTO 9490
9470 PRINT "WRONG - the correct answer is B (PRINT 25+2-3)"
9480 GOTO 9500
9490 PRINT "RIGHT - you've got the right idea about immediate mode"
9500 PRINT
9510 INPUT"press ENTER to continue";T$
9520 GOSUB 9970
9530 PRINT"What command will erase everything in temporary memory?"
9540 PRINT
9550 INPUT"Type in the command using capital letters";T$
9560 PRINT
9570 IF T$ = "NEW" GOTO 9600
9580 PRINT"WRONG - the command is NEW"
9590 GOTO 9610
9600 PRINT"CORRECTO MUNDO - THAT'S RIGHT!"
9610 PRINT
9620 INPUT"press ENTER";T$
9630 GOSUB 9970
9640 PRINT"What command will list all the line numbers and statements"
9650 PRINT"that you have placed in temporary memory?"
9660 PRINT
9670 INPUT"Type in the command using capital letters";T$
9680 PRINT
9690 IF T$ = "LIST" GOTO 9720
9700 PRINT"WRONG - the correct answer is LIST"
9710 GOTO 9730
9720 PRINT"RIGHT YOU ARE!"
9730 PRINT
```

```
9740 INPUT"press ENTER";T$
9750 GOSUB 9970
9760 PRINT"What is the command to delete lines 20 to 50"
9770 PRINT
9780 PRINT"A   DELETE 20-50"
9790 PRINT"B   NEW"
9800 PRINT"C   DELETE 20 to 50"
9810 PRINT"D   ERASE 20 to 50"
9820 PRINT
9830 INPUT"ENTER the correct answer";T$
9840 PRINT
9850 IF T$ = "A" GOTO 9880
9860 PRINT"WRONG the correct answer is A (DELETE 20-50)"
9870 GOTO 9890
9880 PRINT"RIGHT AGAIN!"
9890 PRINT
9900 INPUT"press ENTER";T$
9910 GOSUB 10210
9920 IF T$ = "B" GOTO 8500
9930 RETURN
9940 REM **
9950 REM ** This subroutine clears the screen on any terminal
9960 REM **
9970 FOR X = 1 TO 24
9980    PRINT
9990 NEXT X
10000 RETURN
10010 REM **
10020 REM ** THIS SUBROUTINE IS THE MENU
10030 REM **
10040 PRINT"                    LESSON 1"
10050 PRINT
10060 PRINT"This is the first part of a two part lesson"
10070 PRINT"It is divided into the following sections."
10080 PRINT
10090 PRINT"1)  Introduction          5)  Statements & Programs"
10100 PRINT"2)  Hardware              6)  Print Statement"
10110 PRINT"3)  Software              7)  End & Stop Statement"
10120 PRINT"4)  General Information   8)  Immediate Mode. NEW"
10130 PRINT"                              LIST, DELETE"
10140 PRINT
10150 RETURN
10160 RUN "MENU"
10170 RUN "LESSON1A"
10180 REM **
10190 REM ** THIS SUBROUTINE LETS STUDENT REVIEW SECTIONS AGAIN
```

```
10200 REM **
10210 GOSUB 9970
10220 PRINT"Which do you wish to do"
10230 PRINT
10240 PRINT"A  Continue on"
10250 PRINT"B  Review this section again"
10260 PRINT
10270 INPUT"press the letter opposite your choice and press ENTER";T$
10280 RETURN
```

```
1250 GOSUB 6960
1260 GOSUB 7000
1270 PRINT"A  I'm taking this part in its entirety."
1280 PRINT"B  I wish to review selected areas (or take the test)."
1290 PRINT"C  I want to go to the first part."
1300 PRINT"D  I want to return to the Menu."
1310 PRINT
1320 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
1330 IF T$ = "D" GOTO 7220
1340 IF T$ = "C" GOTO 7230
1350 IF T$ = "B" GOTO 1430
1360 IF T$ <>"A" GOTO 1270
1370 GOSUB 1550
1380 GOSUB 2570
1390 GOSUB 3080
1400 GOSUB 4380
1410 GOSUB 5170
1420 GOSUB 6400
1430 GOSUB 6960
1440 GOSUB 7000
1450 PRINT
1460 PRINT"Please type in the number beside the area you wish"
1470 PRINT"to review (1 through 7) and then press ENTER - press 0 and"
1480 PRINT"press ENTER to return to the Menu."
1490 PRINT
1500 INPUT"What is your choice";N
1510 IF N = 0 GOTO 7220
1520 IF N = 7 GOTO 7210
1530 ON N GOSUB 1550,2570,3080,4380,5170,6400
1540 GOTO 1430
1550 GOSUB 6960
1560 PRINT "                    LIBRARY FUNCTIONS"
1570 PRINT
1580 PRINT"Many mathematical FUNCTIONS such as square root, trignometric"
1590 PRINT"functions, and logritnms are difficult to derive using just"
1600 PRINT"addition, subtraction, multiplication, and division.  To help"
1610 PRINT"us use these FUNCTIONS without deriving them from scratch"
1620 PRINT"each time we want to get a tangent or sine or square root, etc."
1630 PRINT"Microsoft BASIC has a library of commonly used FUNCTIONS"
1640 PRINT"already programmed into permanent memory.  All you have to"
1650 PRINT"do is call them with a BASIC command whenever you want to use"
1660 PRINT"them."
1670 PRINT
1680 PRINT"You identify which function you want to use by using a keyword,"
1690 PRINT"such as SQR for square root."
1700 PRINT
```

```
1710 INPUT"press ENTER";T$
1720 GOSUB 6960
1730 PRINT"                    LIBRARY FUNCTIONS (cont)"
1740 PRINT
1750 PRINT"If you wanted to find the square root of 25, in the IMMEDIATE"
1760 PRINT"mode, you would type in:"
1770 PRINT
1780 PRINT"PRINT SQR(25)"
1790 PRINT
1800 PRINT"Which would result in:"
1810 PRINT
1820 PRINT"5"
1830 PRINT
1940 PRINT"Notice how the keyword precedes the value to be manipulated."
1850 PRINT"and the value is enclosed in parenthesis"
1860 PRINT
1870 PRINT
1880 INPUT"press ENTER";T$
1890 GOSUB 6960
1900 PRINT"                    LIBRARY FUNCTIONS (cont)"
1910 PRINT
1920 PRINT"Another example would be:"
1930 PRINT
1940 PRINT"10 PRINT SQR(2+62)"
1950 PRINT"RUN"
1960 PRINT"Which would give you:"
1970 PRINT
1980 PRINT"8"
1990 PRINT
2000 PRINT"In this example, note that we applied a function to an"
2010 PRINT"expression with more than one term.  This is entirely legal,"
2020 PRINT"and can shorten the number of statements you may need in"
2030 PRINT"your program."
2040 PRINT
2050 INPUT"press ENTER";T$
2060 GOSUB 6960
2070 PRINT"                    LIBRARY FUNCTIONS (cont)"
2080 PRINT
2090 PRINT"You may use a function statement any number of times in your"
2100 PRINT"program.  The different types of LIBRARY FUNCTIONS will be"
2110 PRINT"reviewed in a later lesson."
2120 PRINT
2130 PRINT"If you don't find the function you want in the library, then"
2140 PRINT"you may create your own function.  This is called a"
2150 PRINT"USER DEFINED function.  A USER DEFINED FUNCTION is not"
2160 PRINT"stored permanently in memory, it can only be used in the"
```

```
2170 PRINT"program it was created in.  We will discuss USER DEFINED"
2180 PRINT"FUNCTIONS in a later lesson."
2190 PRINT
2200 PRINT
2210 INPUT"press ENTER";T$
2220 GOSUB 6960
2230 PRINT"QUIZ time!  - USE ONLY CAPITAL LETTERS IN YOUR ANSWERS!"
2240 PRINT
2250 PRINT"Are the library functions stored permanently in memory?"
2260 PRINT
2270 PRINT"A  Yes"
2280 PRINT"B  No"
2290 PRINT
2300 INPUT"press the letter opposite the correct answer and press ENTER";T$
2310 PRINT
2320 IF T$ = "A" THEN GOTO 2350
2330 IF T$ <> "A" THEN PRINT "WRONG - the correct answer is  A (Yes)"
2340 GOTO 2360
2350 PRINT"CORRECT"
2360 PRINT
2370 INPUT"press ENTER to continue";T$
2380 GOSUB 6960
2390 PRINT"If the function for converting a number to an integer is"
2400 PRINT"INT, show the statement for finding the integer value of"
2410 PRINT"27.56.  Use line number 10, and leave only one space"
2420 PRINT"between elements.  Do not include the RUN command."
2430 PRINT
2440 PRINT"DON'T FORGET TO ENCLOSE '27.56' IN PARENTHESIS"
2450 PRINT
2460 INPUT"Type in your answer";T$
2470 PRINT
2480 IF T$ = "10 PRINT INT(27.56)" GOTO 2510
2490 PRINT"WRONG - the correct answer is --- 10 PRINT INT(27.56)"
2500 GOTO 2520
2510 PRINT"CORRECT"
2520 PRINT
2530 INPUT"press ENTER to continue";T$
2540 GOSUB 7120
2550 IF T$ = "B" GOTO 1550
2560 RETURN
2570 GOSUB 6960
2580 PRINT"                        Variables"
2590 PRINT
2600 PRINT"When working with computers, it is necessary to define the type"
2610 PRINT"of data you are manipulating, if for no other reason than to"
2620 PRINT"communicate your program to someone else.  Numbers, such as"
```

```
2630 PRINT"10, 32, 50, 1, etc., are considered CONSTANTS.  Can you guess"
2640 PRINT"why?  Its because they never change, they are always worth a"
2650 PRINT"set amount.   They are CONSTANT."
2660 PRINT
2670 PRINT"On the other hand, in algebra we learned that we could man-"
2680 PRINT"ipulate numbers and define problems easier if we assigned"
2690 PRINT"letters such as X and Y to equations.  In this case, X and Y"
2700 PRINT"are VARIABLES.  That is, they could assume any value we wanted"
2710 PRINT"as long as the value suited the equation."
2720 PRINT
2730 INPUT"press ENTER";T$
2740 GOSUB 6960
2750 PRINT"                    Variables (cont)"
2760 PRINT
2770 PRINT"The way we treated letters in algebra, that is, assigning"
2780 PRINT"them values that were variable and were for calculation"
2790 PRINT"purposes, is the same way we treat them in the computer world."
2900 PRINT
2810 PRINT"For instance, if we give X the value of 8, then the computer"
2820 PRINT"will store the value 8 in a memory location that is labeled X."
2830 PRINT"The value will not change until we assign a new value to the"
2840 PRINT"label X, or quit BASIC."
2850 PRINT
2860 PRINT"There are two fundamental types of variables in BASIC, they"
2870 PRINT"are NUMERIC variables, and STRING variables.  Our previous"
2880 PRINT"example of assigning X a number made it a NUMERIC variable."
2890 PRINT
2900 INPUT"press ENTER";T$
2910 GOSUB 6960
2920 PRINT"If we had assigned a CHARACTER (such as my name, DAN) to a"
2930 PRINT"variable, then we would have created a STRING variable."
2940 PRINT
2950 PRINT"A STRING variable holds data that will not be operated on"
2960 PRINT"mathematically.  (I wouldn't want my name operated on, would"
2970 PRINT"you?)"
2980 PRINT
2990 PRINT"The reason for having STRING variables is so we can do things"
3000 PRINT"like print labels, make word processors, and develop computer"
3010 PRINT"assisted instruction programs.  If these tasks are done in"
3020 PRINT"BASIC, then they are done using STRINGS."
3030 PRINT
3040 INPUT"press ENTER";T$
3050 GOSUB 7120
3060 IF T$ = "B" GOTO 2570
3070 RETURN
3080 GOSUB 6960
```

```
3090 PRINT"                    Numeric Variables"
3100 PRINT
3110 PRINT"In computers we assign values to variables to ease our job:"
3120 PRINT
3130 PRINT"X = 1+2"
3140 PRINT
3150 PRINT"In this case, the value of 3 would be assigned to X and the"
3160 PRINT"computer would store the value in its memory until we either"
3170 PRINT"changed it, or quit BASIC.  In other words, we assigned the"
3180 PRINT"value of 3 to X, but only temporarily.  Take the following"
3190 PRINT"example:"
3200 PRINT
3210 PRINT"10 X = 1+2"
3220 PRINT"20 X = 4"
3230 PRINT
3240 PRINT"What do you think the value of X is if we RUN the example?"
3250 PRINT
3260 INPUT"press ENTER for the answer";T$
3270 GOSUB 6960
3280 PRINT"                    Numeric Variables (cont)"
3290 PRINT
3300 PRINT"Of course, you knew the answer was 4, didn't you?"
3310 PRINT
3320 PRINT"Because long programs sometimes need many variables, Microsoft"
3330 PRINT"BASIC allows you to use all the letters of the alphabet PLUS"
3340 PRINT"it allows you to add a SECOND letter OR number to a variable"
3350 PRINT"to distinguish it from another.  A1, X2, YY, YZ, and FF are"
3360 PRINT"legal variables.  1A, 22, or 3I are not legal.  Can you see"
3370 PRINT"why?  Right, they do not begin with a letter of the alphabet!"
3380 PRINT"(2 letters or 1 letter and 1 number are max length allowed)"
3390 PRINT
3400 PRINT"You may also assign a value to a variable that is assigned to"
3410 PRINT"other variables.  For instance:"
3420 PRINT
3430 INPUT"press ENTER for an example of variable assignment";T$
3440 GOSUB 6960
3450 PRINT"10 X = 4"
3460 PRINT"20 Y = 7"
3470 PRINT"30 Z = X+Y"
3480 PRINT
3490 PRINT"The variable Z is assigned the value of X+Y or 11."
3500 PRINT
3510 PRINT
3520 PRINT"All variables are assigned the value of 0 when you first"
3530 PRINT"start up Microsoft BASIC.  However, some languages assign"
3540 PRINT"indefinite values to all variables at first, and wait for you"
```

```
3550 PRINT"to change them.  That is why you may see programmers setting a"
3560 PRINT"variable to 0 when there appears to be no other reason for it."
3570 PRINT
3580 PRINT
3590 PRINT
3600 INPUT"press ENTER";T$
3610 GOSUB 6960
3620 PRINT"                    Numeric Variables (cont)"
3630 PRINT
3640 PRINT"When you use variables on the right side of an equation"
3650 PRINT"you must have assigned values to the variables previously. It's"
3660 PRINT"a KEY CONCEPT that the equal sign does not mean mathematical"
3670 PRINT"equality.  The equal sign is an ASSIGNMENT statement."
3680 PRINT"It ASSIGNS the value on the right side of the equation"
3690 PRINT"to the variable on the left."
3700 PRINT
3710 PRINT"10 X = X+2"
3720 PRINT
3730 PRINT"In the above statement, X will be assigned the value of "
3740 PRINT"0+2, or 2"
3750 PRINT
3760 INPUT"press ENTER";T$
3770 GOSUB 6960
3780 PRINT"                    Numeric Variables (cont)"
3790 PRINT
3800 PRINT"10 X = 4"
3810 PRINT
3820 PRINT"In the above example, we assigned the value of 4 to X."
3830 PRINT"In some dialects of BASIC, we must use the word LET to assign"
3840 PRINT"a value to a variable."
3850 PRINT
3860 PRINT"10 LET X = 4"
3870 PRINT
3880 PRINT"Such as above.  It is not necessary to use the word LET in"
3890 PRINT"Microsoft BASIC.  We only mention it because you may wish to"
3900 PRINT"copy a program written in another dialect onto Microsoft.  If"
3910 PRINT"you do, you may either leave the LET word in or drop it, the"
3920 PRINT"BASIC language processor will accept either version."
3930 PRINT
3940 INPUT"press ENTER";T$
3950 GOSUB 6960
3960 PRINT"Which of the following is a legal statement in Microsoft BASIC?"
3970 PRINT
3980 PRINT"A  10 LET XY = 2"
3990 PRINT"B  20 1X = 2+3"
4000 PRINT"C  15 4X2 = 56"
```

```
4010 PRINT"D  10 23 = X+Y"
4020 PRINT
4023 INPUT"press the letter opposite the correct answer and press ENTER";T$
4040 PRINT
4050 IF T$ = "A" GOTO 4120
4060 PRINT"WRONG - the correct answer is A ( LET XY = 2)"
4070 PRINT
4080 PRINT"This is a KEY concept. you may wish to review Variables"
4090 PRINT"again, to be sure you understand them."
4100 PRINT
4110 GOTO 4140
4120 PRINT "CORRECT - Good job!"
4130 PRINT
4140 INPUT"press ENTER";T$
4150 GOSUB 6960
4160 PRINT"Which of the following is a legal statement if variables A and"
4170 PRINT"B have previously been assigned a value?"
4180 PRINT
4190 PRINT"A  10 A+B = C"
4200 PRINT"B  10 C = A+B"
4210 PRINT"C  10 22 = A+B"
4220 PRINT"D  10 C22 = A+B"
4230 PRINT
4240 INPUT"press the letter opposite the correct answer";T$
4250 PRINT
4260 IF T$ = "B" GOTO 4320
4270 PRINT"WRONG - the correct answer is B  C = A+B"
4280 PRINT"You may need to review this section if you did not get this"
4290 PRINT"question right."
4300 PRINT
4310 GOTO 4340
4320 PRINT"CORRECT - that was a KEY concept. you're doing good!"
4330 PRINT
4340 INPUT"press ENTER";T$
4350 GOSUB 7120
4360 IF T$ = "B" GOTO 3080
4370 RETURN
4380 GOSUB 6960
4390 PRINT"                    String Variables"
4400 PRINT
4410 PRINT"Whenever you assign CHARACTERS (or NUMBERS that will not be"
4420 PRINT"mathematically manipulated - such as a street address) to a"
4430 PRINT"variable, you have created a STRING variable."
4440 PRINT
4450 PRINT"There is a special way of making a STRING in BASIC.  YOU MUST"
4460 PRINT"attach a dollar sign ($) to the end of a variable label."
```

```
4470 PRINT"When you do, the computer will know that this is a STRING and"
4480 PRINT"will not try to manipulate it.  In addition, everything that"
4490 PRINT"you want to be included in the string must be enclosed in"
4500 PRINT"quotes.  For example:"
4510 PRINT
4520 PRINT"X$ = "CHR$(34)"The author is me"CHR$(34)
4530 PRINT
4540 INPUT"press ENTER";T$
4550 GOSUB 6950
4560 PRINT"                    String Variables (cont)"
4570 PRINT
4580 PRINT"X$ = " CHR$(34)"The author is me"CHR$(34)
4590 PRINT
4600 PRINT"Here, the variable label X is identified as a STRING variable"
4610 PRINT"by the addition of a dollar sign.  Further, the CHARACTER data"
4620 PRINT CHR$(34)"The author is me"CHR$(34)" is assigned to the STRING."
4630 PRINT
4640 PRINT"X1$ = "CHR$(34)"114 West Cottage Street"CHR$(34)
4650 PRINT
4660 PRINT"In this example, we have assigned both letters and numbers to"
4670 PRINT"the STRING."
4680 PRINT
4690 PRINT"Examples of illegal STRING labels would be X, 1X$, X2I$, JIM$"
4700 PRINT
4710 INPUT"press ENTER";T$
4720 GOSUB 6950
4730 PRINT"Which of the following are correct STRINGS?"
4740 PRINT
4750 PRINT"A  AT$ = 12"
4760 PRINT"B  AT$ = "CHR$(34)"12"CHR$(34)
4770 PRINT"C  AT$ = North State Street"
4780 PRINT"D  A$  = North State Street"
4790 PRINT
4800 INPUT"press the letter opposite the correct answer and press ENTER";T$
4810 PRINT
4820 IF T$ = "B" GOTO 4850
4830 PRINT"WRONG - the correct answer is B (AT$ = "CHR$(34)"12"CHR$(34)
4840 GOTO 4860
4850 PRINT"CORRECT - are you sure you are only a student?"
4860 PRINT
4870 INPUT"press ENTER";T$
4880 GOSUB 6950
4890 PRINT"The following program is an example of a BASIC program and its"
4900 PRINT"output:"
4910 PRINT
4920 PRINT"10 T$ = "CHR$(34)"This is really awesome, I mean really,"CHR$(34)
```

```
4930 PRINT"RUN"
4940 PRINT
4950 PRINT"This is really awesome, I mean really."
4960 PRINT
4970 PRINT"Do you see how the computer treated the data?"
4980 PRINT"What would be the output of this program:"
4990 PRINT
5000 PRINT"10 ZZ$ = "CHR$(34)"THIS IS A TEST QUESTION"CHR$(34)
5010 PRINT"RUN"
5020 PRINT
5030 PRINT"Type in the correct answer exactly as it would be printed"
5040 INPUT T$
5050 PRINT
5060 IF T$ = "THIS IS A TEST QUESTION" GOTO 5110
5070 PRINT"WRONG - the correct answer is:"
5080 PRINT
5090 PRINT"THIS IS A TEST QUESTION"
5100 GOTO 5120
5110 PRINT"CORRECT"
5120 PRINT
5130 INPUT"press ENTER";T$
5140 GOSUB 7120
5150 IF T$ = "B" GOTO 4330
5160 RETURN
5170 GOSUB 8950
5180 PRINT"                    Using Arithmetic"
5190 PRINT
5200 PRINT"BASIC will let you use arithmetic to figure out almost any"
5210 PRINT"mathmatical task you would want.  BASIC uses five symbols to"
5220 PRINT"represent addition, subtraction, multiplication, division and"
5230 PRINT"exponentiation (raising something to a power).  Here they are:
5240 PRINT
5250 PRINT"  Symbol            Meaning          Example"
5260 PRINT
5270 PRINT"    +               addition          4+5"
5280 PRINT"    -               subtraction       7-2"
5290 PRINT"    *               multiplication    3*2"
5300 PRINT"                    division          6/2"
5310 PRINT"                    exponentiation    A 2 'A squared)"
5320 PRINT
5330 PRINT" parenthesis ( ) - are also used, just as in algebra)"
5340 PRINT
5350 INPUT"press ENTER";T$
5360 GOSUB 8950
5370 PRINT"                    Using Arithmetic (cont.)"
5380 PRINT
```

```
5390 PRINT"Note that a "CHR$(34)"*"CHR$(34)" always must be used for multiplication"
5400 PRINT"If you tried to use an X or a, the computer"
5410 PRINT"would think you were trying to put in another variable label"
5420 PRINT"and would give you an error message.  Also, you cannot use"
5430 PRINT"terms like 2(5) to mean 2*5, if you do, you will get an error."
5440 PRINT"If you put two variables together, like A and B, to make AB,"
5450 PRINT"you are not multiplying them, rather, you just created a NEW"
5460 PRINT"variable (AB)"
5470 PRINT
5480 PRINT"So always remember to use the asterisk for multiplication."
5490 PRINT
5500 INPUT"press ENTER";T$
5510 GOSUB 5950
5520 PRINT"                    Using Arithmetic (cont)"
5530 PRINT
5540 PRINT"The symbols we just looked at are called ARITHMETIC OPERATORS"
5550 PRINT"and they may be combined in any order in a BASIC statement."
5560 PRINT"However, just like mathematics, the computer will treat some"
5570 PRINT"symbols with a higher priority than others.  For example:"
5580 PRINT
5590 PRINT"X = 10+2.5-(2*3+2^2)"
5600 PRINT
5610 PRINT"In this statement, the computer will scan the line and do"
5620 PRINT"all terms within parentheses first.  Then it"
5630 PRINT"will scan for exponentiation, perform those operations, then"
5640 PRINT"it will scan for multiplication OR division and perform those"
5650 PRINT"operations as it comes to them, and finally, it will scan"
5660 PRINT"for addition OR subtraction and perform those operations."
5670 PRINT
5680 INPUT"press ENTER";T$
5690 GOSUB 5960
5700 PRINT"                    Using Arithmetic (cont)"
5710 PRINT
5720 PRINT"The computer always scans from left to right.  It will scan"
5730 PRINT"once for each category of symbols.  The categories"
5740 PRINT"are restated below."
5750 PRINT
5760 PRINT"     Category                Priority"
5770 PRINT
5780 PRINT"        ^                    HIGHEST"
5790 PRINT"        *                Next HIGHEST"
5800 PRINT"     * or /              Next HIGHEST"
5810 PRINT"     + or -                  LOWEST"
5820 PRINT
5830 INPUT"press ENTER for more";T$
5840 GOSUB 5950
```

```
5850 PRINT"                    Using Arithmetic (cont)"
5860 PRINT
5870 PRINT
5880 PRINT" X = 10+2.5-(2*3+2^2)"
5890 PRINT
5900 PRINT"On the first scan, the computer would do the terms within the"
5910 PRINT"parenthesis.  It would first do exponentiation (2^2 is 4)"
5920 PRINT"and then it would do the multiplication, and finally the addi-"
5930 PRINT"tion.  The value inside the parenthesis would be set at 10"
5940 PRINT"Then it would do the terms outside the parenthesis in order of"
5950 PRINT"importance.  First it would do the division, then it would"
5960 PRINT"do the addition (because it is scanning from left to "
5970 PRINT"right) and the subtraction last.  Finally, it would set the"
5980 PRINT"value of X at 4"
5990 PRINT
6000 INPUT"press ENTER";I$
6010 GOSUB 6960
6020 PRINT"                    Using Arithmetic"
6030 PRINT
6040 PRINT"Parenthesis can be used to establish precedence within a"
6050 PRINT"statement.  Suppose you want to make sure that the LAST part of"
6060 PRINT"a statement is calculated FIRST.  You can use parenthesis."
6070 PRINT"For example:"
6080 PRINT
6090 PRINT"G = 35*(2+1)"
6100 PRINT
6110 PRINT"is much different than:"
6120 PRINT
6130 PRINT"G=35*2+1"
6140 PRINT
6150 PRINT"Do you see why?  The first value assigned to G is 35*3, the"
6160 PRINT"second value assigned is 1090.  Study the example carefully."
6170 INPUT"press ENTER";I$
6180 GOSUB 6960
6190 PRINT"Which of the following statements will assign the value of 10"
6200 PRINT"to the variable M"
6210 PRINT
6220 PRINT"A  M = 1+2(2^2)-2"
6230 PRINT"B  M = 5*2+(3*3+1)-10"
6240 PRINT"C  M = 20/2(3*3+1)"
6250 PRINT"D  M = 10*3-2"
6260 PRINT
6270 INPUT"press the letter opposite the correct answer and press ENTER";I$
6280 PRINT
6290 IF I$ = "B" GOTO 6240
6300 PRINT"WRONG - the right answer is B  M = 5*2+(3*3+1)-10"
```

```
6310 PRINT"                              (M = 5*2+(10)-10)"
6320 PRINT"                              (M = 10 + 10 - 10.)"
6330 GOTO 6350
6340 PRINT"CORRECT"
6350 PRINT
6360 INPUT"press ENTER to continue";T$
6370 GOSUB 7120
6380 IF T$ = "B" GOTO 5170
6390 RETURN
6400 GOSUB 6960
6410 PRINT"                        Comparing Variables"
6420 PRINT
6430 PRINT"BASIC uses symbols to compare values to determine relationships"
6440 PRINT"such as whether one variable is less than, more than, or equal"
6450 PRINT"to another variable.  We have already used one of these symbols"
6460 PRINT"it is called the equal sign (=).  When you start programming,"
6470 PRINT"you will often want to check to see if one variable is dif-"
6480 PRINT"ferent than another.   There are six symbols you can use to"
6490 PRINT"do this."
6500 PRINT
6510 INPUT"press ENTER for examples";T$
6520 GOSUB 6960
6530 PRINT"                   Comparing Variables (cont)"
6540 PRINT
6550 PRINT"                        (symbol table)"
6560 PRINT
6570 PRINT
6580 PRINT"    Symbol           Meaning          Example"
6590 PRINT
6600 PRINT"      =               equal            A=2"
6610 PRINT"      <               less than        A<B"
6620 PRINT"      >               greater than     A>B"
6630 PRINT"      <>              not equal to     A<>B"
6640 PRINT"     <=            less than or equal  A<=B"
6650 PRINT"     >=         greater than or equal  A>=B"
6660 PRINT
6670 PRINT"We will discuss these in more detail in a later lesson."
6680 PRINT
6690 INPUT"press ENTER";T$
6700 GOSUB 6960
6710 PRINT"That concludes this lesson.  When you hit ENTER you will"
6720 PRINT"be returned to the start of this part.  You may either take"
6730 PRINT"the test or review selected areas."
6740 PRINT
6750 PRINT"By now, you should understand the following program."
6760 PRINT
```

```
6770 PRINT"10 PRINT"CHR$(34)"The product of 10 times 8 is "CHR$(34)"
6780 PRINT"20 PRINT 10 * 8"
6790 PRINT"30 PRINT"
6800 PRINT"40 PRINT"
6810 PRINT"50 END"
6820 PRINT"RUN"
6830 PRINT
6840 PRINT"Your homework assignment will require you to write a program"
6850 PRINT"similar to this. Line 10 prints the string, line 20 prints"
6860 PRINT"the mathematical calculation. Lines 30 and 40"
6870 PRINT"print two blank lines.  You make the program work by using"
6880 PRINT"the RUN word after you have entered the statements."
6890 PRINT"Your actual homework assignment is at the end of the test."
6900 PRINT
6910 INPUT"press ENTER";T$
6920 RUN
6930 REM **
6940 REM ** This subroutine clears the screen on any terminal
6950 REM **
6960 FOR X = 1 TO 24
6970    PRINT
6980 NEXT X
6990 RETURN
7000 PRINT"                    LESSON 1B"
7010 PRINT
7020 PRINT"This is the second part of a two part lesson"
7030 PRINT"It is divided into the following sections."
7040 PRINT
7050 PRINT"1)  Library Functions     4)  String Variables"
7060 PRINT"2)  Variables (general)   5)  Using Arithmetic"
7070 PRINT"3)  Numeric Variables     6)  Comparing Variables"
7080 PRINT"                              & Lesson Summary"
7090 PRINT"                  7)  TEST
7100 PRINT
7110 RETURN
7120 GOSUB 6960
7130 PRINT"Which do you want to do?"
7140 PRINT
7150 PRINT"A  Continue on"
7160 PRINT"B  Review this lesson again"
7170 PRINT
7180 INPUT"press the letter opposite your choice and press ENTER";T$
7190 IF T$ <> "A" AND T$ <> "B" GOTO 7190
7200 RETURN
7210 RUN "TEST1"
7220 RUN "MENU"
```

***** Listing of Program 'LESSON1A' ***** 07/10/83 - 01:59:45

```
7230 RUN "LESSON1"
7240 END
```

***** Listing of Program 'TEST1' ***** 07/10/83 - 02:17:03

```
1000 REM **
1010 REM ** LESSON:  TEST1                VERSION: 1 AUG 83
1020 REM ** AUTHOR:  CAPT DAN CREAGAN
1030 REM **          AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM **            N$(X) = NAMES ARRAY, USED TO READ IN SEQ-
1070 REM **                    UENTIAL NAMES, AND TO WRITE OUT
1080 REM **                    UPDATE NAMES.
1090 REM **            S(X)  = SCORES ARRAY - USED TO READ AND
1100 REM **                    WRITE SCORES
1110 REM **            Q(X)  = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM **                    CORRECT ANSWERS.  IF AN ARRAY
1130 REM **                    ELEMENT EQUALS 1, THE ANSWER WAS
1140 REM **                    CORRECT
1150 REM **
1160 CLEAR 3000
1170 GOSUB 4130
1180 DIM N$(1000)
1190 DIM Q(10)
1200 DIM S(1000)
1210 PRINT"                    FINAL TEST (lesson 1)"
1220 PRINT
1230 PRINT"This test consists of 10 questions, you must get 70 percent"
1240 PRINT"of them correct to pass.  (that's 7 right out of the 10 ques-"
1250 PRINT"tions).  Use only capital letters in your answers, don't"
1260 PRINT"include extra spaces or letters.  If you answer a question wrong,"
1270 PRINT"you get the correct answer, plus a reference for review."
1280 PRINT"In addition, you will get a synopsis of areas for review"
1290 PRINT"at the end of the test."
1300 PRINT
1310 PRINT"If you successfully pass the test, you will be given your"
1320 PRINT"homework assignment.  GOOD LUCK!"
1330 PRINT
1340 INPUT"press ENTER to continue";T$
1350 GOSUB 4130
1360 PRINT"Is a computer program called Software?"
1370 PRINT
1380 PRINT"A   Yes"
1390 PRINT"B   No"
1400 PRINT
1410 INPUT"press the letter opposite the correct answer and press ENTER";T$
1420 PRINT
1430 IF T$ = "A" THEN GOTO 1470
1440 PRINT"WRONG - the correct answer is A (Yes, programs are software)"
1450 PRINT"        review part 1, hardware and software."
```

```
1460 GOTO 1490
1470 PRINT"CORRECT"
1480 Q(1) = 1
1490 PRINT
1500 INPUT"press ENTER";T$
1510 GOSUB 4130
1520 PRINT"Which of the following is an example of a FUNCTION"
1530 PRINT
1540 PRINT"A  LIST"
1550 PRINT"B  NEW"
1560 PRINT"C  SQR"
1570 PRINT"D  ADD"
1580 PRINT
1590 INPUT"press the letter opposite the correct answer and press ENTER";T$
1600 PRINT
1610 IF T$ = "C" THEN GOTO 1650
1620 PRINT"WRONG - the correct answer is C (SQR)"
1621 PRINT"        LIST will list your program lines, NEW erases your"
1622 PRINT"        program, and ADD is not a legal BASIC word."
1630 PRINT"        review part 2, FUNCTIONS"
1640 GOTO 1670
1650 PRINT"CORRECT"
1660 Q(2) = 1
1670 PRINT
1680 INPUT"press ENTER to continue";T$
1690 GOSUB 4130
1700 PRINT"Which statement would print the word TEST"
1710 PRINT
1720 PRINT"A  PRINT TEST"
1730 PRINT"B  PRINT "CHR$(34)"TEST"CHR$(34)
1740 PRINT"C  OUTPUT "CHR$(34)"TEST"CHR$(34)
1750 PRINT"D  PRINT 'TEST'"
1760 PRINT
1770 INPUT"press the letter opposite the correct answer and press ENTER";T$
1780 PRINT
1790 IF T$ = "B" THEN GOTO 1830
1800 PRINT"WRONG - the correct answer is B (PRINT "CHR$(34)"TEST"CHR$(34)")"
1802 PRINT"        Answer A would treat the word TEST like a variable."
1804 PRINT"        answer C has an illegal BASIC word (output) and"
1806 PRINT"        answer D uses the wrong characters for quotes."
1810 PRINT"        review part 1, PRINT, and part 2, String Variables"
1820 GOTO 1950
1830 PRINT"CORRECT"
1840 Q(3) = 1
1850 PRINT
1860 INPUT"press ENTER";T$
```

```
1870 GOSUB 4130
1880 PRINT"What would the following program's output be?"
1890 PRINT
1900 PRINT"10 X = SQR(4)"
1910 PRINT"20 PRINT 5^X"
1920 PRINT"RUN"
1930 PRINT
1940 INPUT"Type in your answer and press ENTER";T$
1950 PRINT
1960 IF T$ = "25" THEN GOTO 2000
1970 PRINT "WRONG  -  the correct answer is 25"
1972 PRINT"          line 10 puts the square root of 4 into the"
1974 PRINT"          variable X, line 20 causes 5 to be taken to"
1976 PRINT"          the power of 2. 5 squared is 25."
1980 PRINT"          review part 1, PRINT, and part 2. FUNCTIONS"
1990 GOTO 2020
2000 PRINT"CORRECT"
2010 Q(4) = 1
2020 PRINT
2030 INPUT"press ENTER";T$
2040 GOSUB 4130
2050 PRINT"Give the necessary statement to print a blank line.  Use"
2060 PRINT"line number 10 and leave one blank space between terms."
2070 PRINT
2080 INPUT"What's your answer";T$
2090 PRINT
2100 IF T$ = "10 PRINT" THEN GOTO 2140
2110 PRINT"WRONG - the correct answer is 10 PRINT"
2120 PRINT"        review part 1, PRINT"
2130 GOTO 2160
2140 PRINT"CORRECT"
2150 Q(5) = 1
2160 PRINT
2170 INPUT"press ENTER";T$
2180 GOSUB 4130
2190 PRINT"The two types of FUNCTIONS are LIBRARY and COMPUTER, TRUE or"
2200 PRINT"FALSE?"
2210 PRINT
2220 PRINT"A   TRUE"
2230 PRINT"B   FALSE"
2240 PRINT
2250 INPUT"press the letter opposite the correct answer and press ENTER";T$
2260 PRINT
2270 IF T$ = "B" THEN GOTO 2330
2280 PRINT"WRONG - the correct answer is B (False) - the two types"
2290 PRINT"        of functions are LIBRARY and USER.  Review part"
```

```
2300 PRINT"       2. FUNCTIONS."
2310 PRINT
2320 GOTO 2350
2330 PRINT"CORRECT"
2340 Q(6) = 1
2350 PRINT
2360 INPUT"press ENTER";T$
2370 GOSUB 4130
2380 PRINT"Which of the following statements is invalid?"
2390 PRINT
2400 PRINT"A  25 = X"
2410 PRINT"B  PRINT SQR(25)"
2420 PRINT"C  PRINT 25**10"
2430 PRINT"D  L = M+N"
2440 PRINT
2450 INPUT"Type in the letter opposite the INCORRECT statement";T$
2460 PRINT
2470 IF T$ = "A" THEN GOTO 2510
2480 PRINT"WRONG - the INCORRECT statement is A (25 = X)"
2482 PRINT"       You cannot set a constant (25) equal to a variable"
2490 PRINT"       review part 1, PRINT, part 2, FUNCTIONs"
2500 GOTO 2530
2510 PRINT"CORRECT"
2520 Q(7) = 1
2530 PRINT
2540 INPUT"press ENTER";T$
2550 GOSUB 4130
2560 PRINT"A string variable is made of mathematical equations which"
2570 PRINT"will be manipulated by the computer and saved in temporary"
2580 PRINT"memory.    TRUE or FALSE?"
2590 PRINT
2600 INPUT"Type in TRUE or type in FALSE for this statement";T$
2610 PRINT
2620 IF T$ = "FALSE" THEN GOTO 2670
2630 PRINT"WRONG  -  the answer is FALSE"
2640 PRINT"          strings are not used for manipulating math"
2650 PRINT"          equations. Review part 2, Strings."
2660 GOTO 2690
2670 PRINT"CORRECT"
2680 Q(8) = 1
2690 PRINT
2700 INPUT"press ENTER";T$
2710 GOSUB 4130
2720 PRINT"Which of the following is an example of a peripheral device?"
2730 PRINT
2740 PRINT"A  CPU"
```

```
2750 PRINT"B  MEMORY"
2760 PRINT"C  KEYBOARD"
2770 PRINT
2780 INPUT"press the letter opposite the correct answer and press ENTER";T$
2790 PRINT
2800 IF T$ = "C"  THEN GOTO 2840
2810 PRINT"WRONG - the correct answer is  C (keyboard)"
2812 PRINT"        Answers A & B are not peripherals, they are"
2814 PRINT"        part of the main computer structure."
2820 PRINT"        review part 1, General Information"
2830 GOTO 2860
2840 PRINT"CORRECT"
2850 Q(9) = 1
2860 PRINT
2870 INPUT"press ENTER";T$
2880 GOSUB 4130
2890 PRINT"If you had the following program in memory:"
2900 PRINT
2910 PRINT"10  G=34"
2920 PRINT"20  X=40"
2930 PRINT"30  Y=10"
2940 PRINT"40  L=3"
2950 PRINT
2960 PRINT"What statement would you use to remove the middle two lines?"
2970 PRINT
2980 PRINT"A  NEW"
2990 PRINT"B  PRINT 10 + 40"
3000 PRINT"C  DELETE 20 to 30"
3010 PRINT"D  DELETE 20-30"
3020 PRINT
3030 INPUT"press the letter opposite the correct answer and press ENTER";T$
3040 PRINT
3050 IF T$ = "D" THEN GOTO 3090
3060 PRINT"WRONG - the correct answer is D (DELETE 20-30)"
3062 PRINT"        Answer A would delete the whole program, and"
3064 PRINT"        answer B prints the sum of 10 and 40. Answer C"
3066 PRINT"        would cause a syntax error."
3070 PRINT"        review part 1, IMMEDIATE, NEW, DELETE"
3080 GOTO 3110
3090 PRINT"CORRECT"
3100 Q(10) = 1
3110 PRINT
3120 INPUT"press ENTER";T$
3130 GOSUB 4130
3140 FOR X = 1 TO 10
3150    Y = Y+Q(X)
```

```
3160 NEXT X
3170 PRINT"You have finished the test, out of 10 possible correct answers"
3180 PRINT"you scored "Y"."
3190 PRINT
3200 IF Y > 6 THEN  PRINT"YOU HAVE PASSED"
3210 GOSUB 3950
3220 IF Y > 6 THEN GOTO 3300
3230 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
3240 PRINT
3250 PRINT"YOU SHOULD RETAKE LESSON 1 BEFORE GOING FARTHER!"
3260 PRINT
3270 PRINT"You will be returned to the Menu."
3280 PRINT
3290 GOTO 4170
3300 PRINT
3310 PRINT"Do you want your score recorded on a permanent file?"
3320 PRINT
3330 PRINT"A  YES"
3340 PRINT"B  NO"
3350 PRINT
3360 INPUT"Which";T$
3370 IF T$ = "B" THEN GOTO 3650
3380 GOSUB 4130
3390 PRINT"To record your score, we must open a file and put your name"
3400 PRINT"in it.  Therefore, surprisingly, we need your name.  If your"
3410 PRINT"name is not unique among the students likely to take this test,"
3420 PRINT"please contact your test monitor for an identifying word that"
3430 PRINT"will make you unique.  Then enter that word below."
3440 PRINT
3450 PRINT"If you have already entered a score previously, be sure to"
3460 PRINT"enter the same name you used before.  (use all capitals)"
3470 PRINT
3480 INPUT"ENTER your word or name now";T$
3490 OPEN"I",1,"SCORE1"
3500 X = 0
3510 IF EOF(1) THEN GOTO 3570
3520 X = X+1
3530 INPUT#1,N$(X)
3540 INPUT#1,S(X)
3550 IF N$(X) = T$ THEN GOTO 3810
3560 GOTO 3510
3570 CLOSE
3580 X = X+1
3590 N$(X) = T$
3600 S(X) = Y
3610 OPEN"O",1,"SCORE1"
```

```
3620 FOR W = 1 TO X
3630   PRINT#1,N$(W)
3640   PRINT#1,S(W)
3650 NEXT W
3660 GOSUB 4170
3670 PRINT"You are now qualified to go to LESSON 2."
3680 PRINT
3690 PRINT"Your homework assignment is:"
3700 PRINT
3710 PRINT"Write a short program that will state the following when RUN:"
3712 PRINT
3720 PRINT"THE SUM OF 3, 2, AND 22 IS"
3730 PRINT"27"
3732 PRINT
3740 PRINT
3742 PRINT"Make the second statement ('27') actually calculate the"
3744 PRINT"sum of 3, 2, and 22.  - similar to the example at the"
3746 PRINT"end of the last part of lesson 1.  Finally, print two"
3748 PRINT"blank lines at the end of the program."
3750 PRINT"Be sure to copy this instruction down before you go on."
3800 GOTO 4170
3810 S(X) = Y
3820 IF EOF(1) THEN CLOSE:GOTO 3860
3830 X = X+1
3840 INPUT#1, N$(X), S(X)
3850 GOTO 3820
3860 OPEN"O",1,"SCORE1"
3870   FOR W = 1 TO X
3880       PRINT#1,N$(W)
3890       PRINT#1,S(W)
3900   NEXT W
3910 PRINT
3920 PRINT"You may now take LESSON 2.  You will be returned to the MENU"
3930 PRINT"from where you may go to LESSON 2 or quit."
3940 GOTO 4170
3950 IF Y=10 THEN RETURN
3960 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
3970 PRINT
3980 IF Q(1) = 0 THEN PRINT"    part 1, Hardware and Software"
3990 IF Q(2) = 0 OR Q(4) = 0 OR Q(6) = 0 THEN PRINT"    part 2, Functions"
4000 IF Q(3) = 0 THEN PRINT"    part 1, Print, and part 2, String Variables"
4010 IF Q(5) = 0 THEN PRINT"    part 1, Print"
4020 IF Q(7) = 0 THEN PRINT"    part 2, Variables"
4030 IF Q(8) = 0 THEN PRINT"    part 2, String Variables"
4040 IF Q(9) = 0 THEN PRINT"    part 1, General Information"
4050 IF Q(10) = 0 THEN PRINT"    part 1, IMMEDIATE"
```

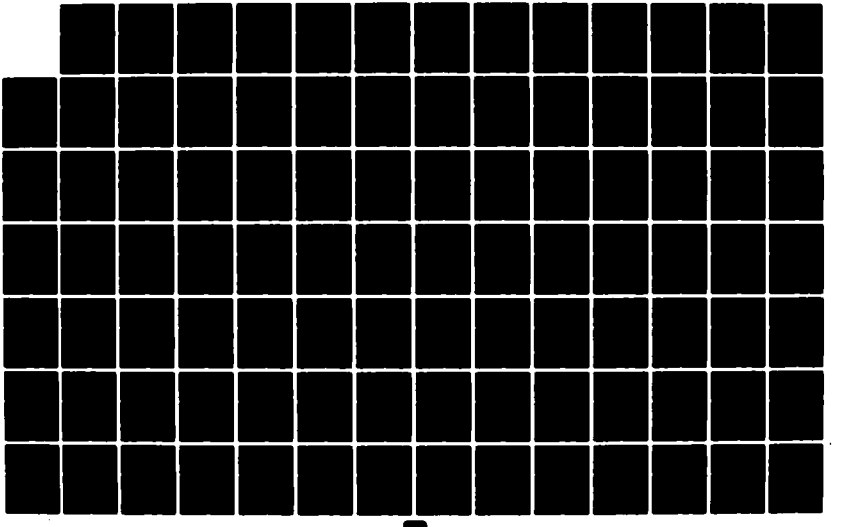
```
4060 PRINT
4070 INPUT"press ENTER";T$
4080 GOSUB 4130
4090 RETURN
4100 REM **
4110 REM ** this subroutine clears the screen"
4120 REM **
4130 FOR X = 1 TO 24
4140     PRINT
4150 NEXT X
4160 RETURN
4170 PRINT
4180 INPUT"press ENTER to return to MENU";T$
4190 RUN"MENU"
```

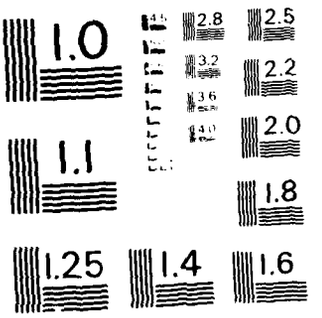
```
1000 REM ** THIS PROGRAM STARTED ON 20 APRIL 1983
1010 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
1020 REM ** TITLE: LESSON 2
1030 REM **
1040 REM **
1050 REM **
1060 REM **
1070 GOSUB 8120
1080 PRINT"LESSON:  BASIC 2            VERSION: 1 AUGUST 83
1090 PRINT
1100 PRINT"TIME REQUIRED TO COMPLETE LESSON:  About One Hour"
1110 PRINT
1120 PRINT
1130 PRINT"AUTHOR:  Capt Danny J. Creagan"
1140 PRINT"         Air Force Institute of Technology"
1150 PRINT
1160 PRINT"OBJECTIVE:  To teach the student about permanent storage,"
1170 PRINT"            how to handle data in BASIC, and how to branch to"
1180 PRINT"            different parts of a program."
1190 PRINT
1200 PRINT
1210 PRINT
1220 PRINT
1230 INPUT"press the ENTER key to continue";T$
1240 GOSUB 8120
1250 GOSUB 8160
1260 PRINT"A  I'm taking this part in its entirety."
1270 PRINT"B  I wish to review selected areas."
1280 PRINT"C  I want to go to the second part."
1290 PRINT"D  I want to return to the Menu."
1300 PRINT
1310 INPUT"Press either  capital A, B, C, or D and then press ENTER";T$
1320 IF T$ = "D" F  . 1360
1330 IF T$ = "C" GOTO 9370
1340 IF T$ = "B" GOTO 1430
1350 IF T$ <>"A" GOTO 1260
1360 GOSUB 1540
1370 GOSUB 1950
1380 GOSUB 3870
1390 GOSUB 5000
1400 GOSUB 5670
1410 GOSUB 8120
1420 GOTO 9370
1430 GOSUB 8120
1440 GOSUB 8160
1450 PRINT
```

```
1460 PRINT"Please type in the number beside the area you wish"
1470 PRINT"to review (1 through 5) and then press ENTER - press 0 and"
1480 PRINT"press ENTER to return to the Menu."
1490 PRINT
1500 INPUT"What is your choice";N
1510 IF N = 0 GOTO 9360
1520 ON N GOSUB 1540,1950,3870,5000,5670
1530 GOTO 1430
1540 GOSUB 8120
1550 INPUT"Do you wish to see an answer to the homework problem (Y/N)";T$
1560 IF LEFT$(T$,1) = "N" OR LEFT$(T$,1) = "n" THEN GOTO 1660
1570 GOSUB 8120
1580 PRINT"Here is one possible way to complete your homework:"
1590 PRINT
1600 PRINT"10 PRINT"CHR$(34)"THE SUM OF 3, 2, AND 22 IS"CHR$(34)
1610 PRINT"20 PRINT 3 + 2 + 22"
1620 PRINT"30 PRINT"
1622 PRINT"40 PRINT"
1623 PRINT"RUN"
1624 PRINT"THE SUM OF 3, 2, AND 22 IS"
1625 PRINT"27"
1626 PRINT
1630 PRINT
1640 INPUT"You can try this one if you had trouble with yours.  Press ENTER";T$
1650 T$ = ""
1660 GOSUB 8120
1670 PRINT"                         Introduction"
1680 PRINT
1690 PRINT "In this lesson we will cover some of the most exciting and"
1700 PRINT "useful commands in the BASIC language.  When we left off,"
1710 PRINT "in lesson 1, we had discussed some of the fundamental commands"
1720 PRINT "that you must use just to get BASIC started.  Now, we will"
1730 PRINT "discover how to SAVE our programs for future use, how to"
1740 PRINT"control data input in our program, and how to leave little"
1750 PRINT"messages in our program so that other programmers can under-"
1760 PRINT"stand what we are trying to do.  Most importantly, we will"
1770 PRINT"discover how to branch to different parts of a program"
1780 PRINT"depending on our data manipulation requirements.  That way, one"
1790 PRINT"program can be extremely flexible and do many different kinds"
1800 PRINT"of work for us."
1810 PRINT
1820 INPUT"press ENTER to continue";T$
1830 GOSUB 8120
1840 PRINT"                    Introduction"
1850 PRINT
1860 PRINT"After taking this lesson, we recommend you practice some of the"
```

AD-A134 386 COMPUTER ASSISTED INSTRUCTION IN BASIC(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS D J CREAGAN 28 SEP 83 AFIT-LSSR-29-83 2/6

UNCLASSIFIED F/G 9/2 NL





```
1870 PRINT"techniques you have learned.  The best way is to write a"
1880 PRINT"short program of your own and get it working.  Then get a"
1890 PRINT"short program from a magazine or book and type it in.  Don't"
1900 PRINT"be afraid to experiment with it.  The best way of learning"
1910 PRINT"BASIC is to practice it."
1920 PRINT
1930 INPUT"press ENTER";T$
1940 RETURN
1950 GOSUB 8120
1960 PRINT"                    Filenames"
1970 PRINT
1980 PRINT"Remember, in lesson 1, when we discovered how to make programs"
1990 PRINT"that could be RUN over and over?  We said then that the pro-"
2000 PRINT"gram was stored in TEMPORARY memory.  If you tried a few of"
2010 PRINT"the examples that were given, you will have noticed that the"
2020 PRINT"program was destroyed whenever you left BASIC.  This section"
2030 PRINT"and the next section will show you how to SAVE a program, and"
2040 PRINT"then call it back from PERMANENT storage.  That way, when "
2050 PRINT"you've spent hours making the best data manager ever written,"
2060 PRINT"you won't have to re-write it when you turn on the machine"
2070 PRINT"again!"
2080 PRINT
2090 PRINT"PERMANENT storage is the way we store data for an indefinite"
2100 PRINT"period.  We usually use DISKS or TAPE for PERMANENT storage."
2110 INPUT"press ENTER";T$
2120 GOSUB 8120
2130 PRINT"                    Filenames (cont)"
2140 PRINT
2150 PRINT"For the purposes of this lesson, we will assume you only use"
2160 PRINT"DISKS for permanent storage."
2170 PRINT
2180 PRINT"A DISK is a platter of iron-oxide coated material that stores"
2190 PRINT"data almost the same way that an audio tape stores music."
2200 PRINT"A DISK comes in many sizes and with many different storage"
2210 PRINT"capabilities.  Fortunately, the way we store data on disk when"
2220 PRINT"we are using Microsoft BASIC is standardized for almost all"
2230 PRINT"installations.  (there is a slight difference if you are using"
2240 PRINT"a TRS-80, we will explain it as we go along)"
2250 PRINT
2260 INPUT"press ENTER";T$
2270 GOSUB 8120
2280 PRINT"                    Filenames (cont)"
2290 PRINT
2300 PRINT"What happens when you store data?  Well, the computer takes"
2310 PRINT"care of most of the details, it waits until you tell it to"
2320 PRINT"store a program, then it searches the available storage areas"
```

```
2330 PRINT"to see if there is room for storage of your masterpiece, then"
2340 PRINT"it writes your data on the DISK.  Remember, there may be MANY"
2350 PRINT"programs stored on a disk; therefore, each program must have"
2360 PRINT"a label that distinguishes it from the others. That way, the"
2370 PRINT"computer can find your program when you ask for it again."
2380 PRINT
2390 INPUT"press ENTER";T$
2400 GOSUB 8120
2410 PRINT"                    Filenames (cont)"
2420 PRINT
2430 PRINT"This label is called a FILENAME.  FILENAMES are very strictly"
2440 PRINT"controlled by the computer.  They must follow the following"
2450 PRINT"format EXACTLY."
2460 PRINT
2470 PRINT"         TRS-80              CROMEMCO (or CPM)"
2480 PRINT
2490 PRINT"   (filename)/(extension)     (filename).(extension)"
2500 PRINT
2510 PRINT"Notice that the only difference between a TRS-80 and Cromemco"
2520 PRINT"is that the TRS-80 has a slash, '/', between the filename"
2530 PRINT"and the extension, while the Cromemco has a period or dot, '.'"
2540 PRINT
2550 INPUT"press ENTER";T$
2560 GOSUB 8120
2570 PRINT"                    Filenames (cont)"
2580 PRINT
2590 PRINT"      TRS-80                 CROMEMCO"
2600 PRINT
2610 PRINT"(filename)/(extension)     (filename).(extension)"
2620 PRINT
2630 PRINT"In the two examples, (filename), is an alphabetical character"
2640 PRINT"string no longer than eight (8) characters. (extension) is "
2650 PRINT"a file extension name that is also an alphabetical character"
2660 PRINT"string.  The extension must not be longer than three (3)"
2670 PRINT"characters.  The extension is OPTIONAL but, if used, must"
2680 PRINT"follow the format EXACTLY.  NUMBERS may be used in both "
2690 PRINT"filenames and extensions, but they must NOT be the FIRST letter"
2700 PRINT
2710 INPUT"press ENTER";T$
2720 GOSUB 8120
2730 PRINT"                    Filenames (cont)"
2740 PRINT"Here are some examples of legal filenames for your computer"
2750 PRINT
2760 PRINT"   MYPROG/BAS              MYPROG.BAS"
2770 PRINT"   MRHAPPY/BAS             MRHAPPY.BAS"
2780 PRINT"   SWIMFIN2/BAS             SWIMFIN"
```

```
2790 PRINT"    GOODNESS                    GOODNESS.BAS"
2800 PRINT
2810 PRINT"Notice that the filenames do not have to make sense, just so"
2820 PRINT"they mean something to the programmer who made them. It"
2830 PRINT"would be unwise to name your program something common,"
2840 PRINT"like TEST.BAS, because someone else has probably already used"
2850 PRINT"that name.  If they have, you will destroy their program when"
2860 PRINT"you SAVE your program to disk.  Each program name must be"
2870 PRINT"unique."
2880 INPUT"press ENTER";T$
2890 GOSUB 8120
2900 PRINT"                    Filenames (cont)"
2910 PRINT
2920 PRINT"       SWIMFIN.BAS"
2930 PRINT
2940 PRINT"Notice, in the above filename, the extension is BAS.  This"
2950 PRINT"would normally indicate that the file is a BASIC file (you "
2960 PRINT"may have word processor files, machine language files, or"
2970 PRINT"a variety of others).  A good tip is to always save your"
2980 PRINT"BASIC files with this extension.  That way, when you read"
2990 PRINT"the disk directory, you can tell that you have to go to BASIC"
3000 PRINT"to run any program that has the extension - .BAS."
3010 PRINT
3020 INPUT"press ENTER";T$
3030 GOSUB 8120
3040 PRINT"                    Filenames (cont)"
3050 PRINT
3060 PRINT"This section is very important to you.  It has shown you what"
3070 PRINT"filenames are and what legal filenames look like.  In the"
3080 PRINT"future you will use them a lot.  Be sure you under-"
3090 PRINT"stand the idea behind filenames before you continue.  It"
3100 PRINT"would be a good idea to look them up in your operating"
3110 PRINT"manual for Microsoft BASIC.  There are many rules that were not"
3120 PRINT"covered here, but the rules we covered will get you by for now."
3130 PRINT
3140 INPUT"press ENTER";T$
3150 GOSUB 8120
3160 INPUT"ENTER a 'T' if using a TRS-80, or 'C' if CPM or CROMEMCO";T$
3170 IF T$ = "T" GOTO 3540
3180 GOSUB 8120
3190 PRINT"In the Cromemco system, which of the following would be"
3200 PRINT"considered a legal filename."
3210 PRINT
3220 PRINT"A   XXXXXXXX.FIL"
3230 PRINT"B   24lesson.BAS"
3240 PRINT"C   TEST-BAS"
```

```
3250 PRINT"D   LUNCHTALK"
3260 PRINT
3270 INPUT"press the letter opposite the correct answer and press ENTER";T$
3280 IF T$ = "A" GOTO 3330
3290 PRINT
3300 PRINT"WRONG the correct answer is A (XXXXXXXX.FIL)"
3310 PRINT
3320 GOTO 3350
3330 PRINT"CORRECT - good job!"
3340 PRINT
3350 INPUT"press ENTER";T$
3360 GOSUB 9120
3370 PRINT"Is the extension necessary for a filename to be legal?"
3380 PRINT
3390 PRINT"A  YES"
3400 PRINT"B  NO"
3410 PRINT
3420 INPUT"press the letter opposite the correct answer and press ENTER";T$
3430 PRINT
3440 IF T$ = "B" GOTO 3470
3450 PRINT"WRONG - the correct answer is B (the extension is not needed)"
3460 GOTO 3480
3470 PRINT"CORRECT"
3480 PRINT
3490 INPUT"press ENTER";T$
3500 GOSUB 9120
3510 GOSUB 8270
3520 IF T$ = "B" GOTO 1950
3530 RETURN
3540 GOSUB 9120
3550 PRINT"which of the following filenames is correct"
3560 PRINT
3570 PRINT"A  XXXXXXXX/BAS"
3580 PRINT"B  24lesson.BAS"
3590 PRINT"C  TEST-BAS"
3600 PRINT"D  LUNCHTALK"
3610 PRINT
3620 INPUT"press the letter opposite the correct answer and press ENTER";T$
3630 PRINT
3640 IF T$ = "A" GOTO 3670
3650 PRINT"WRONG the correct answer is A (XXXXXXXX/BAS)"
3660 GOTO 3680
3670 PRINT"CORRECT"
3680 PRINT
3690 INPUT"press ENTER";T$
3700 GOSUB 912
```

```
3710 PRINT"An extension to a filename is mandatory."
3720 PRINT
3730 PRINT"A  YES"
3740 PRINT"B  NO"
3750 PRINT
3760 INPUT"press the letter opposite the correct answer and press ENTER";T$
3770 PRINT
3780 IF T$ = "B" GOTO 3810
3790 PRINT"WRONG - the correct answer is B (an extension is not needed)"
3800 GOTO 3820
3810 PRINT"CORRECT"
3820 PRINT
3830 INPUT"press ENTER";T$
3840 GOSUB 8270
3850 IF T$ = "B" GOTO 1950
3860 RETURN
3870 GOSUB 8120
3880 PRINT"                    SAVE, LOAD and RUN"
3890 PRINT
3900 PRINT"At the beginning of the last section, we said we would discover"
3910 PRINT"how to SAVE our programs so we wouldn't have to keep typing"
3920 PRINT"them in all the time.  Well, this is it.  To SAVE your program,"
3930 PRINT"(let's say you called it MYPROG.BAS), all you do is:"
3940 PRINT
3950 PRINT"            1) Type in the program"
3960 PRINT"            2) Type SAVE "CHR$(34)"MYPROG.BAS"CHR$(34)
3970 PRINT"            3) Congratulate yourself on a good job!"
3980 PRINT
3990 PRINT"Be sure to notice that the filename is enclosed in quotation"
4000 PRINT"marks.  That is mandatory, if you don't enclose the name in"
4010 PRINT"quotes, the command will 'BOMB' (it will fail)."
4020 PRINT
4030 INPUT"press ENTER";T$
4040 GOSUB 8120
4050 PRINT"                    SAVE, LOAD, and RUN (cont)"
4060 PRINT
4070 PRINT"There are other things you should be aware of before you try"
4080 PRINT"to SAVE a program.  First, there should be enough room on the"
4090 PRINT"disk to hold the program.  If you are using a CROMEMCO hard"
4100 PRINT"disk, you will probably not have any problem in this area.  ASK"
4110 PRINT"YOUR SYSTEM OPERATOR for more information.  If you are using"
4120 PRINT"TRS-80 small floppy disk, then go to the COMMAND mode by"
4130 PRINT"typing in CMD"CHR$(34)"S"CHR$(34)" and then typing in DIR :0 or"
4140 PRINT"DIR :1.  (drive 1 is called :0 and drive 2 is called :1), then"
4150 PRINT"watch the display.  You will see the free space left on the disk"
4160 PRINT and a DIRECTORY of the files on the disk.  If you have ever"
```

```
4170 PRINT"30 grams, then you have enough room for almost any program."
4180 PRINT"SEE YOUR OPERATING MANUAL FOR MORE DETAILS."
4190 PRINT
4200 INPUT"press ENTER";T$
4210 GOSUB 8120
4220 PRINT"                    SAVE, RUN, and LOAD (cont)"
4230 PRINT
4240 PRINT"Remember, if you have enough space, then just type in this:"
4250 PRINT
4260 PRINT"          SAVE "CHR$(34)"MYPROG.BAS"CHR$(34)" or "CHR$(34)"MYPROG.BAS"CHR$(34)
4270 PRINT"               if using a TRS-80)"
4280 PRINT
4290 PRINT"Now that we know how to SAVE a program, how do we get it back"
4300 PRINT"so we can RUN it again?  That's easy.  Just type in:"
4310 PRINT
4320 PRINT"          RUN"CHR$(34)"MYPROG.BAS"CHR$(34)
4330 PRINT
4340 INPUT"press ENTER";T$
4350 GOSUB 8120
4360 PRINT"                    SAVE, RUN, and LOAD (cont)"
4370 PRINT
4380 PRINT"           RUN"CHR$(34)"MYPROG.BAS"CHR$(34)
4390 PRINT
4400 PRINT"When you type in the command, the computer will load your"
4410 PRINT"program THAT YOU HAD PREVIOUSLY SAVED, and RUN it."
4420 PRINT
4430 PRINT"What if you just saved a piece of a program because you were
4440 PRINT"tired, expecting to come back at a later date and add to it?
4450 PRINT"If you did that, then you wouldn't want to RUN the program.
4460 PRINT"You would just want to LOAD the program and LIST it to be
4470 PRINT"sure it was the right one, then add the line numbers you need"
4480 PRINT"to complete the program."
4490 PRINT
4500 INPUT"press ENTER";T$
4510 GOSUB 8120
4520 PRINT"                    SAVE, RUN, and LOAD (cont)"
4530 PRINT
4540 PRINT"You would LOAD the program using the same format as for SAVEing"
4550 PRINT"and RUNning it.  That is:"
4560 PRINT
4570 PRINT"          LOAD"CHR$(34)"MYPROG.BAS"CHR$(34)
4580 PRINT
4590 PRINT"BE SURE TO SAVE THE PROGRAM AGAIN AFTER YOU MODIFY IT, BECAUSE"
4600 PRINT"ONLY A COPY OF THE OLD VERSION WILL BE ON THE DISK!"
4610 PRINT
4620 INPUT"press ENTER";T$
```

```
4630 GOSUB 8120
4640 PRINT"Which of the following is the correct command to SAVE a"
4650 PRINT"program named FRITZ"
4660 PRINT
4670 PRINT"A  SAVE FRITZ"
4680 PRINT"B  LOAD FRITZ.BAS"
4690 PRINT"C  SAVE MYPROG"
4700 PRINT"D  SAVE"CHR$(34)"FRITZ"CHR$(34)
4710 PRINT
4720 INPUT"press the letter opposite the correct answer and press ENTER";T$
4730 PRINT
4740 IF T$ = "D" GOTO 4770
4750 PRINT"WRONG - the correct answer is D (SAVE"CHR$(34)"FRITZ"CHR$(34)")"
4760 GOTO 4780
4770 PRINT"CORRECT - FRITZ thanks you!"
4780 PRINT
4790 INPUT"press ENTER";T$
4800 GOSUB 8120
4810 PRINT"You have just gotten to BASIC and want to load a program you"
4820 PRINT"have been working on.  The filename for the program is SQUEEZE."
4830 PRINT"How would you get the program from permanent storage to "
4840 PRINT"temporary memory?  Which of the following would you type in:"
4850 PRINT
4860 PRINT"A  SQUEEZE"
4870 PRINT"B  LOAD"CHR$(34)"SQUEEZE"CHR$(34)
4880 PRINT"C  LOAD"CHR$(34)"SQUEEZE.BAS"CHR$(34)
4890 PRINT
4900 INPUT"press the letter opposite the correct answer and press ENTER";T$
4910 PRINT
4920 IF T$ = "B" GOTO 4950
4930 PRINT"WRONG - the correct answer is B (LOAD"CHR$(34)"SQUEEZE"CHR$(34)
4940 PRINT:GOTO 4960
4950 PRINT"CORRECT - give yourself a HUG!":PRINT
4960 INPUT"press ENTER";T$
4970 GOSUB 8270
4980 IF T$ = "B" GOTO 3870
4990 RETURN
5000 GOSUB 8120
5010 PRINT"                          REMarks"
5020 PRINT
5030 PRINT"Something we should start early in our programming life, is"
5040 PRINT"DOCUMENTATION of how a program runs.  You can include state-"
5050 PRINT"ments within a program THAT WILL NOT BE TOUCHED BY THE COMPUTER"
5060 PRINT"and will add to the clarity of your program.  That way, when"
5070 PRINT"you LIST your program (or when another programmer does), you"
5080 PRINT"can read the reminders left behind and more fully understand"
```

```
5090 PRINT"the program."
5100 PRINT
5110 PRINT"Those statements are called REM statements (REMark statements)."
5120 PRINT
5130 INPUT"press ENTER";T$
5140 GOSUB 8120
5150 PRINT"                    REMarks (cont)"
5160 PRINT
5170 PRINT"The format for a REM statement is  (LINE #) REM (REMARKS)"
5180 PRINT
5190 PRINT"An example is:"
5200 PRINT
5210 PRINT"10 REM This is now a REMark statement is made"
5220 PRINT
5230 PRINT"When the computer sees REM it ignores all data that follows"
5240 PRINT
5250 INPUT"press ENTER";T$
5260 GOSUB 8120
5270 PRINT"Here is an example of REMarks in a program:"
5280 PRINT
5290 PRINT"5  R = 2"
5300 PRINT"10  REM  the variable X in the next line is approx equal to PI"
5310 PRINT"20  X = 3.14"
5320 PRINT"25  C = X*R**2"
5330 PRINT"30  PRINT C"
5340 PRINT"RUN"
5350 PRINT
5360 PRINT"Which would give:"
5370 PRINT
5380 PRINT"12.56"
5390 PRINT
5400 PRINT"Notice that the REM statement was not printed"
5410 PRINT
5420 INPUT"press ENTER";T$
5430 GOSUB 8120
5440 PRINT"10  PRINT "CHR$(34)"SWEETUMS is my girl."CHR$(34)
5450 PRINT"20  REM   "CHR$(34)"SWEETUMS is my girl."CHR$(34)
5460 PRINT"30  PRINT "CHR$(34)"(her real name is CANDY)"CHR$(34)
5470 PRINT"40  REM   "CHR$(34)"I love her"CHR$(34)
5480 PRINT"RUN"
5490 PRINT
5500 PRINT"The above program would print the follwing (TRUE OR FALSE):"
5510 PRINT
5520 PRINT"SWEETUMS is my girl."
5530 PRINT"(her real name is CANDY)"
5540 PRINT
```

```
5550 INPUT"Type in TRUE or FALSE, whichever is correct";T$
5560 PRINT
5570 IF T$ = "TRUE" GOTO 5610
5580 PRINT"WRONG - the correct answer is TRUE - REM statements are not"
5590 PRINT"printed!"
5600 GOTO 5620
5610 PRINT"CORRECT - good job!"
5620 PRINT
5630 INPUT"press ENTER";T$
5640 GOSUB 8270
5650 IF T$ = "B" GOTO 5000
5660 RETURN
5670 GOSUB 8120
5680 PRINT"                    INPUT Statements"
5690 PRINT
5700 PRINT"We saw in the first lesson that DATA can be assigned to a"
5710 PRINT"variable using the equals '=' sign.  For example:"
5720 PRINT
5730 PRINT"10  X = 10"
5740 PRINT"20  PRINT X"
5750 PRINT"RUN"
5760 PRINT
5770 PRINT"Gives us:"
5780 PRINT
5790 PRINT"10"
5800 PRINT
5810 PRINT"In this example, we assigned 10 to X in line number 10."
5820 PRINT
5830 INPUT"press ENTER";T$
5840 GOSUB 8120
5850 PRINT"                    INPUT Statements (cont)"
5860 PRINT
5870 PRINT"It is also possible to assign data while the program is run-"
5880 PRINT"ning'  THAT IS HOW THIS PROGRAM ASKS YOU QUESTIONS.  It then"
5890 PRINT"tests your answer to see if you were right."
5900 PRINT
5910 PRINT"The BASIC word that it uses to ask the question is called an"
5920 PRINT"INPUT Statement.  It looks like this:"
5930 PRINT
5940 PRINT"10  INPUT"CHR$(34)"press the correct letter, then press ENTER"CHR$(34)";T$"
5950 PRINT
5960 INPUT"press ENTER";T$
5970 GOSUB 8120
5980 PRINT"                    INPUT Statements (cont)"
5990 PRINT
6000 PRINT"10 INPUT"CHR$(34)"press the correct letter, then press ENTER"CHR$(34)";T$
```

```
6010 PRINT"RUN"
6020 PRINT
6030 PRINT"Gives us:"
6040 PRINT
6050 PRINT"press the correct letter, then press ENTER?"
6060 PRINT
6070 PRINT"Notice that a question mark is automatically inserted after the"
6080 PRINT"message is printed.  When the question is answered, the letter"
6090 PRINT"that the student selects is assigned to T$, just as if we"
6100 PRINT"had assigned a value to it in an equals statement."
6110 PRINT"Also note that a semi-colon is placed after the text'"
6120 PRINT
6130 INPUT"press ENTER";T$
6140 GOSUB 8120
6150 PRINT"                    INPUT Statements (cont)"
6160 PRINT
6170 PRINT"Here is another example:"
6180 PRINT
6190 PRINT"10  A = 20"
6200 PRINT"20  INPUT"CHR$(34)"Enter a number between 1 and 9"CHR$(34)";N
6210 PRINT"30  C = A*N"
6220 PRINT"RUN"
6230 PRINT
6240 PRINT"Gives us:"
6250 PRINT"Enter a number between 1 and 9?"
6260 PRINT"(if we ENTER a 5 then)"
6270 PRINT
6280 PRINT"100"
6290 PRINT
6300 INPUT"press ENTER";T$
6310 GOSUB 8120
6320 PRINT"                    INPUT Statements (cont)"
6330 PRINT
6340 PRINT"10  INPUT"CHR$(34)"ENTER a number between 1 and 9"CHR$(34)";N"
6350 PRINT
6360 PRINT"We have learned then, that the INPUT statement allows you"
6370 PRINT"to ENTER data in a program while it is running.  It does"
6380 PRINT"this by stopping the program and waiting for you to enter"
6390 PRINT"data.  When you do, it sets the data equal to the variable"
6400 PRINT"on the end of the INPUT statement."
6410 PRINT
6420 PRINT"Between the message or prompt and the variable,"
6430 PRINT"you must place a semi-colon (look at example above)."
6440 PRINT
6450 INPUT"press ENTER";T$
6460 GOSUB 8120
```

```
6470 PRINT"                    INPUT Statements"
6480 PRINT
6490 PRINT"You may use the INPUT statement without using a prompt or text"
6500 PRINT"message.  If you do, then you must NOT put in a semi-colon."
6510 PRINT"For example:"
6520 PRINT
6530 PRINT"10  PRINT"CHR$(34)"When you see a question mark, ENTER a 5"CHR$(34)
6540 PRINT"20  INPUT N"
6550 PRINT"RUN"
6560 PRINT
6570 PRINT"Gives you:"
6580 PRINT
6590 PRINT"When you see a question mark, ENTER a 5"
6600 PRINT"?"
6610 PRINT
6620 INPUT"press ENTER";T$
6630 GOSUB 8120
6640 PRINT"                    INPUT Statements (cont)":PRINT
6650 PRINT"10 PRINT"CHR$(34)"When you see a question mark, ENTER a 5"CHR$(34)
6660 PRINT"20 INPUT N"
6670 PRINT"RUN"
6680 PRINT
6690 PRINT"When you see a question mark, ENTER a 5"
6700 PRINT"?"
6710 PRINT
6720 PRINT"Notice how the INPUT statement prompt (question mark) is on"
6730 PRINT"the following line?  If we hadn't included line 10, we wouldn't"
6740 PRINT"know what to do when we saw the question mark.  That's why you"
6750 PRINT"will see the text included in an INPUT statement most of the"
6760 PRINT"time.  However, both ways are used."
6770 PRINT
6780 INPUT"press ENTER";T$
6790 GOSUB 8120
6800 PRINT"                    INPUT Statement (cont)"
6810 PRINT
6820 PRINT"10  INPUT"CHR$(34)"ENTER a number between 1 and 9"CHR$(34)";N"
6830 PRINT"RUN"
6840 PRINT
6850 PRINT"If we were to save this example, and run it at a later date,"
6860 PRINT"We would always be asked for a number between 1 and 9.  'N'"
6870 PRINT"would always be changed from zero to the number we give it."
6880 PRINT
6890 PRINT"The values we assign to variables using the INPUT statement are"
6900 PRINT"not stored as part of the program.  They are only temporarily"
6910 PRINT"held until we leave BASIC. They are reset to zero when we rerun"
6920 PRINT"the program."
```

```
6930 PRINT
6940 INPUT"press ENTER";T$
6950 GOSUB 8120
6980 PRINT"                    INPUT Statements (cont)"
6970 PRINT
6990 PRINT"We have seen examples of both STRING variables and NUMERIC"
6990 PRINT"variables and we have seen both used with INPUT statements."
7000 PRINT
7010 PRINT"If you try to ENTER string data into a NUMERIC variable, you"
7020 PRINT"will get a '?REDO' message.  That means you tried to ENTER data"
7030 PRINT"that was not proper for a NUMERIC variable."
7040 PRINT
7050 PRINT"A problem develops when you think you are entering NUMERIC data"
7060 PRINT"and you ENTER it into a STRING.  You will not get an error"
7070 PRINT"message.  Remember, put the right kind of variable on the end"
7080 PRINT"of the INPUT statement (you will be tested on this)."
7090 PRINT
7100 INPUT"press ENTER";T$
7110 GOSUB 8120
7120 PRINT"                    INPUT Statements (cont)"
7130 PRINT
7140 PRINT"You may ENTER data into more than one variable using only ONE"
7150 PRINT"INPUT statement.  Just put a comma ',' between the variables,"
7160 PRINT"and a question mark will be prompted for each variable."
7170 PRINT"For example:"
7180 PRINT
7190 PRINT"10 INPUT"CHR$(34)"ENTER three (3) numbers"CHR$(34)";A,B,C"
7200 PRINT"RUN"
7210 PRINT
7220 PRINT"ENTER three (3) numbers? 10"
7230 PRINT"?? 22"
7240 PRINT"?? 5"
7250 PRINT
7260 INPUT"press ENTER";T$
7270 GOSUB 8120
7280 PRINT"                    INPUT Statements (cont)"
7290 PRINT
7300 PRINT"10 INPUT"CHR$(34)"ENTER three (3) numbers"CHR$(34)";A,B,C"
7310 PRINT"RUN"
7320 PRINT
7330 PRINT"ENTER three (3) numbers? 10"
7340 PRINT"?? 22"
7350 PRINT"?? 5"
7360 PRINT
7370 PRINT"Notice how the computer keeps prompting you with question marks"
7380 PRINT"until it gets all of its data?  This can be a very useful"
```

```
7390 PRINT"routine, especially when you are asking for coordinates or"
7400 PRINT"for some other paired data input."
7410 PRINT
7420 INPUT"press ENTER";T$
7430 GOSUB 8120
7440 PRINT"                    INPUT Statements (cont)"
7450 PRINT
7460 PRINT"10  INPUT"CHR$(34)"ENTER three (3) numbers"CHR$(34)";A,B,C"
7470 PRINT"RUN"
7480 PRINT
7490 PRINT"ENTER three (3) numbers? 10,22.5"
7500 PRINT
7510 PRINT"Notice that we hastened our data input, in this case, by"
7520 PRINT"ENTERing all the data on one line, separated by commas."
7530 PRINT
7540 PRINT"You may choose either way of ENTERing the data, it makes no"
7550 PRINT"difference."
7560 PRINT
7570 PRINT
7580 INPUT"press ENTER";T$
7590 GOSUB 8120
7600 PRINT"An INPUT statement CAN assign a value to a variable while the"
7610 PRINT"program is running? (TRUE or FALSE)"
7620 PRINT
7630 INPUT"ENTER the word TRUE or ENTER the word FALSE";T$
7640 PRINT
7650 IF T$ = "TRUE" GOTO 7690
7660 PRINT"WRONG - an INPUT statement IS used for inputting data while"
7670 PRINT"the program is running"
7680 GOTO 7700
7690 PRINT"CORRECT"
7700 PRINT
7710 INPUT"press ENTER";T$
7720 GOSUB 8120
7730 PRINT"What is the prompt that an INPUT statement ALWAYS gives?"
7740 PRINT
7750 PRINT"A  A question mark"
7760 PRINT"B  Two question marks"
7770 PRINT"C  Quotes"
7780 PRINT"D  the word INPUT?"
7790 PRINT
7800 INPUT"ENTER the correct letter (either A,B,C, or D)";T$
7810 PRINT
7820 IF T$ = "A" GOTO 7850
7830 PRINT"WRONG - the prompt that is ALWAYS given is a question mark"
7840 GOTO 7860
```

```
7850 PRINT"CORRECT - GREAT!"
7860 PRINT
7870 INPUT"press ENTER";T$
7880 GOSUB 8120
7890 PRINT"Which of the following is a valid response to this statement:"
7900 PRINT
7910 PRINT"20 INPUT A,B,C$"
7920 PRINT
7930 PRINT"A  2,ten,15"
7940 PRINT"B  2,10,15 North Ela"
7950 PRINT"C  TWO,1,22"
7960 PRINT"D  22,NONE,8"
7970 PRINT
7980 INPUT"ENTER the letter opposite the correct response and press ENTER";T$
7990 PRINT
8000 IF T$ = "B" GOTO 8030
8010 PRINT"WRONG - the correct answer is B (2,10,15 North Ela)"
8020 GOTO 8040
8030 PRINT"CORRECT - good, this section is just about done"
8040 PRINT
8050 INPUT"press ENTER";T$
8060 GOSUB 8270
8070 IF T$ = "B" GOTO 5670
8080 RETURN
8090 REM **
8100 REM ** This subroutine clears the screen on any terminal
8110 REM **
8120 FOR X = 1 TO 24
8130    PRINT
8140 NEXT X
8150 RETURN
8160 PRINT"                    LESSON 2"
8170 PRINT
8180 PRINT"This is the first part of a two part lesson"
8190 PRINT"It is divided into the following sections."
8200 PRINT
8210 PRINT"1)  Introduction          4)  REMarks"
8220 PRINT"2)  Filenames             5)  INPUT Statements"
8230 PRINT"3)  SAVE, LOAD, RUN"
8240 PRINT
8250 PRINT
8260 RETURN
8270 GOSUB 8120
8280 PRINT"Which do you wish to do?"
8290 PRINT
8300 PRINT"A  Continue on"
```

```
8310 PRINT"B  Review this section again"
8320 PRINT
8330 INPUT"press the letter opposite the correct answer and press ENTER";T$
8340 IF T$ = "A" OR T$ = "B" THEN RETURN
8350 GOTO 8270
8360 RUN "MENU"
8370 PRINT"GOING TO SECOND PART - PLEASE STANDBY"
8372 RUN"LESSON2A"
8380 END
```

```
260 GOSUB 6950
270 GOSUB 6990
280 PRINT"A  I'm taking this part in its entirety."
290 PRINT"B  I wish to review selected areas (or take the test)."
300 PRINT"C  I want to go to the first part."
310 PRINT"D  I want to return to the Menu."
320 PRINT
330 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
340 IF T$ = "D" GOTO 7690
350 IF T$ = "C" GOTO 7700
360 IF T$ = "B" GOTO 430
370 IF T$ <>"A" GOTO 280
380 GOSUB 550
390 GOSUB 3900
400 GOSUB 4720
410 GOSUB 6000
420 GOSUB 6640
430 GOSUB 6950
440 GOSUB 6990
450 PRINT
460 PRINT"Please type in the number beside the area you wish"
470 PRINT"to review (1 through 5) and then press ENTER - press 0 and"
480 PRINT"press ENTER to return to the Menu."
490 PRINT
500 INPUT"What is your choice";N
510 IF N = 0 THEN GOTO 7690
520 IF N = 5 THEN GOTO 7680
530 ON N GOSUB 550,3900,4720,6000,6630
540 GOTO 430
550 GOSUB 6950
560 PRINT"                    READ and DATA Statements"
570 PRINT
580 PRINT"In the first part of this lesson, we learned that the INPUT"
590 PRINT"statement is very efficient for assigning data to a variable"
600 PRINT"while the program is running.  However, when we have many"
610 PRINT"data points to assign to variables, we need a more effi-"
620 PRINT"cient mode.  Think of how tedious it would be if you had"
630 PRINT"to write statements to assign 250 data points in a program!"
640 PRINT"(It is not unusual to have 10 times 250 data points for large"
650 PRINT"regressions or forecasting programs!)"
660 PRINT
670 PRINT"How do we handle such a huge workload?  One way is to use"
680 PRINT"READ and DATA statements"
690 PRINT
700 INPUT"press ENTER";T$
710 GOSUB 6950
```

```
720 PRINT"                    READ and DATA (cont)"
730 PRINT
740 PRINT"READ and DATA statements are used like the equals sign '=' is"
750 PRINT"used, but they are much faster and more versatile.  Also,"
760 PRINT"READ and DATA statements OPERATE WITHIN THE PROGRAM INSTEAD"
770 PRINT"OF INTERFACING YOU WITH THE PROGRAM."
780 PRINT
790 PRINT"READ and DATA are two separate statements, but they are ALWAYS"
800 PRINT"used WITH each other.  The READ statement assigns the data"
810 PRINT"as the program runs, and the DATA statement holds the values"
820 PRINT"to be assigned."
830 PRINT
840 INPUT"press ENTER";T$
850 GOSUB 6950
860 PRINT"                    READ and DATA (cont)"
870 PRINT
880 PRINT"The format of the READ statement is:"
890 PRINT
900 PRINT"(line number) READ (variable or variables)"
910 PRINT
920 PRINT"An example of a READ statement that would read values into"
930 PRINT"variables B1, A, and XX$ is:"
940 PRINT
950 PRINT"10 READ B1,A,XX$"
960 PRINT
970 INPUT"press ENTER";T$
980 GOSUB 6950
990 PRINT"                    READ and DATA (cont)"
1000 PRINT
1010 PRINT"10 READ B1,A,XX$"
1020 PRINT
1030 PRINT"Note that both numeric and string variables may be 'read'"
1050 PRINT"An example of a DATA statement that would be read is:"
1060 PRINT
1070 PRINT"20 DATA 22,15,"CHR$(34)"AIN'T she sweet?"CHR$(34)
1080 PRINT
1090 PRINT"Notice that the DATA statement has a different line number, but"
1100 PRINT"it follows the same format as the READ statement.  When these"
1110 PRINT"two statements are placed in a program, the variables B1, A, "
1120 PRINT"and XX$ would hold 22, 15, and "CHR$(34)"AIN'T she sweet?"CHR$(34)" respectively."
1130 PRINT
1140 INPUT"press ENTER";T$
1150 GOSUB 6950
1160 PRINT"                    READ and DATA (cont)"
1170 PRINT
1180 PRINT"The two statements can appear anywhere in a program and in any"
```

```
1190 PRINT"order, but for clarity, programmers usually place the DATA "
1200 PRINT"statement after the READ statement.  Also, the DATA statements"
1210 PRINT"are usually grouped with other DATA statements in the program."
1220 PRINT"We do that because it makes it easier to figure out another"
1230 PRINT"program when there is an order to how the programmer entered"
1240 PRINT"his statements."
1250 PRINT
1260 INPUT"press ENTER";T$
1270 GOSUB 6950
1280 PRINT"Is the following statement TRUE or FALSE?"
1290 PRINT
1300 PRINT"A READ statement reads values from a DATA statement."
1310 PRINT"and places the values in variables that are to the"
1320 PRINT"right of the READ word."
1330 PRINT
1340 INPUT"ENTER the word TRUE or ENTER the word FALSE";T$
1350 PRINT
1360 IF T$ = "TRUE" THEN GOTO 1390
1370 PRINT"WRONG - the sentence is valid."
1380 GOTO 1400
1390 PRINT"CORRECT"
1400 PRINT
1410 INPUT"press ENTER";T$
1420 GOSUB 6950
1430 PRINT"                    READ and DATA (cont)"
1440 PRINT
1450 PRINT"10 READ B1,A,XX$"
1460 PRINT
1470 PRINT"20 DATA 22.5,"CHR$(34)"AIN'T she sweet?"CHR$(34)
1480 PRINT
1490 PRINT"The variables in the READ statement, and the values in the "
1500 PRINT"DATA statement are separated by commas, and the string is"
1510 PRINT"enclosed in quotes.  You cannot assign string data to a "
1520 PRINT"numeric variable, if you do, your computer will throw up'"
1530 PRINT"(well, at the very least it will ALWAYS give you an ERROR"
1540 PRINT"message)"
1550 PRINT
1560 INPUT"press ENTER";T$
1570 GOSUB 6950
1580 PRINT"                    READ and DATA (cont)"
1590 PRINT"10 READ B1,A Y"
1600 PRINT"20 PRINT B1,A,Y"
1610 PRINT"30 PRINT "CHR$(34)"The average of these numbers is"CHR$(34)";"
1620 PRINT"40 PRINT (B1+A+Y)/3"
1630 PRINT"50 DATA 5.10.15"
1640 PRINT"RUN"
```

```
1650 PRINT
1660 PRINT"Gives us:"
1670 PRINT
1680 PRINT"5      10      15"
1690 PRINT"The average of these numbers is 10"
1700 PRINT
1710 PRINT"WOW!  Study this example for a moment.  Note that the calcula-"
1720 PRINT"tions were printed beside the message instead of below it."
1730 PRINT
1740 INPUT"press ENTER";T$
1750 GOSUB 6950
1760 PRINT"                    READ and DATA (cont)"
1770 PRINT
1780 PRINT"The calculations were not printed on another line because of"
1790 PRINT"the semi-colon after the print statement in line 30."
1800 PRINT
1810 PRINT"10 READ B1,A,Y"
1820 PRINT"20 PRINT B1,A,Y"
1830 PRINT"30 PRINT "CHR$(34)"The average of these numbers is"CHR$(34)";"
1840 PRINT"40 PRINT B1,A,Y"
1850 PRINT"50 DATA 5,10,15"
1860 PRINT
1870 PRINT"Line 10 read the data in line 50, line 20 PRINTed it, and"
1880 PRINT"line 30 printed the message.  The calculations in line"
1890 PRINT"40 were printed on the end of the message due to the semi-colon"
1900 PRINT"at the tail of line 30."
1910 PRINT
1920 INPUT"press ENTER";T$
1930 GOSUB 6950
1940 PRINT"                    READ and DATA (cont)"
1950 PRINT
1960 PRINT"Let's look at it once more:"
1970 PRINT
1980 PRINT"10 READ B1,A,Y"
1990 PRINT"20 PRINT B1,A,Y"
2000 PRINT"30 PRINT "CHR$(34)"The average of these numbers is"CHR$(34)";"
2010 PRINT"40 PRINT (B1+A+Y)/3"
2020 PRINT"50 DATA 5,10,15"
2030 PRINT
2040 PRINT"Gave us:"
2050 PRINT
2060 PRINT"5      10      15"
2070 PRINT"The average of these numbers is 10"
2080 PRINT
2090 INPUT"press ENTER";T$
2100 GOSUB 6950
```

```
2110 PRINT"Now for a neat example:"
2120 PRINT
2130 PRINT"10 READ A1,B1,C1"
2140 PRINT"20 PRINT A1,B1,C1"
2150 PRINT"30 DATA 1,2"
2160 PRINT"RUN"
2170 PRINT
2180 PRINT"Gives us:"
2190 PRINT
2200 PRINT"OUT OF DATA IN 10"
2210 PRINT
2220 PRINT"The BASIC language processor gave us an ERROR message that"
2230 PRINT"indicates we didn't have enough data for the number of varia-"
2240 PRINT"bles that we tried to READ."
2250 PRINT
2260 INPUT"press ENTER";T$
2270 GOSUB 6950
2280 PRINT"                    READ and DATA (cont)"
2290 PRINT
2300 PRINT"10 READ A1,B1,C1"
2310 PRINT"20 PRINT A1,B1,C1"
2320 PRINT"30 DATA 1,2"
2330 PRINT
2340 PRINT"This program will BOMB (fail) because it will try to find a"
2350 PRINT"non-existent data point for the variable C1.  If there had"
2360 PRINT"been more DATA points than READ variables, the program would"
2370 PRINT"have worked just fine.  The next frame has an example of this."
2380 PRINT
2390 INPUT"press ENTER";T$
2400 GOSUB 6950
2410 PRINT"                    READ and DATA (cont)"
2420 PRINT
2430 PRINT"10 READ A1,B1"
2440 PRINT"20 PRINT A1,B1"
2450 PRINT"30 DATA 1,2,3,4,99"
2460 PRINT
2470 PRINT"Gives us:"
2480 PRINT
2490 PRINT"1       2"
2500 PRINT
2510 PRINT
2520 PRINT"Although there were more data points that could have been read,"
2530 PRINT"there were no more variables left to READ them, so the program"
2540 PRINT"stopped.  This would not cause an ERROR message."
2550 PRINT
2560 INPUT"press ENTER";T$
```

```
2570 GOSUB 6950
2580 PRINT"Is the following program valid?"
2590 PRINT
2600 PRINT"10 READ X,Y,Z"
2610 PRINT"20 DATA 25,2,15,55,64"
2620 PRINT"30 PRINT Z"
2630 PRINT
2640 PRINT"A  Yes"
2650 PRINT"B  No"
2660 PRINT
2670 INPUT"Enter the letter opposite the correct answer";T$
2680 PRINT
2690 IF T$ = "A" THEN GOTO 2720
2700 PRINT"WRONG - the program is valid"
2710 GOTO 2730
2720 PRINT"CORRECT !"
2730 PRINT
2740 INPUT"press ENTER";T$
2750 PRINT
2760 INPUT"What is the value of Z that will be printed out";T$
2770 PRINT
2780 IF T$ = "15" THEN GOTO 2930
2790 PRINT"WRONG - the correct answer is 15, Z is the third"
2800 PRINT"        variable to be read, so the third data point"
2810 PRINT"        is put in it."
2820 GOTO 2840
2830 PRINT"CORRECT !"
2840 PRINT
2850 INPUT"press ENTER";T$
2860 GOSUB 6950
2870 PRINT"                    READ and DATA (cont)"
2880 PRINT
2890 PRINT"Suppose you want to READ the same data points into DIFFERENT"
2900 PRINT"variables?  Or perhaps you are making a program that will deal"
2910 PRINT"a deck of cards out, and you want to start over when you get"
2920 PRINT"to 52.  You can reset the DATA statements so that variables"
2930 PRINT"will be assigned old data points by using the RESTORE"
2940 PRINT"statement."
2950 PRINT
2960 INPUT"press ENTER";T$
2970 GOSUB 6950
2980 PRINT"The RESTORE statement resets the DATA statements.  After a"
2990 PRINT"RESTORE command, the next variable that is READ will be"
3000 PRINT"assigned the value that is just after the first DATA word."
3010 PRINT
3020 PRINT"10 READ A1,B1"
```

```
3030 PRINT"20 PRINT A1,B1"
3040 PRINT"30 RESTORE
3050 PRINT"40 READ C1,D1"
3060 PRINT"50 PRINT C1,D1"
3070 PRINT"60 DATA 1,2"
3080 PRINT"RUN"
3090 PRINT
3100 PRINT"1      2"
3110 PRINT"1      2"
3120 PRINT
3130 INPUT"Study this example closely and then press ENTER";T$
3140 GOSUB 6950
3150 PRINT"                    READ and DATA (cont)"
3160 PRINT
3170 PRINT"This has been an extra long section and, if you are new to"
3180 PRINT"BASIC, you probably are confused about some of the rules."
3190 PRINT"Don't let that worry you.  Get out your BASIC manual (it came"
3200 PRINT"with your computer), and, after the little quiz that is"
3210 PRINT"coming up, go back and review this section again.  THEN"
3220 PRINT"PRACTICE the techniques once you are through with this "
3230 PRINT"lesson.  It is very important that you start practicing what"
3240 PRINT"you are learning.  Practice with the manual beside you, and"
3250 PRINT"don't be afraid to ask an experienced programmer when you are"
3260 PRINT"confused.  Use the examples seen in this program, or make up"
3270 PRINT"your own.  You are one step closer to being a BASIC programmer!"
3280 PRINT
3290 INPUT"press ENTER";T$
3300 GOSUB 6950
3310 PRINT"Is the following statement TRUE or FALSE?"
3320 PRINT
3330 PRINT"The RESTORE statement causes the READ/DATA combination"
3340 PRINT"to reset to the first data point to the right of the"
3350 PRINT"first DATA statement."
3360 PRINT
3370 PRINT"A  TRUE"
3380 PRINT"B  FALSE"
3390 PRINT
3400 INPUT"ENTER the letter opposite the correct answer";T$
3410 PRINT
3420 IF T$ = "A" THEN GOTO 3460
3430 PRINT"WRONG - the RESTORE command DOES reset the READ/DATA pair"
3440 PRINT"        to the first data point past the first DATA word."
3450 GOTO 3470
3460 PRINT"CORRECT - SUPER !"
3470 PRINT
3480 INPUT"press ENTER";T$
```

```
3490 GOSUB 6950
3500 PRINT"READ and DATA statements are used within a program, and they"
3510 PRINT"do NOT stop the program so DATA can be entered."
3520 PRINT
3530 PRINT"A  TRUE"
3540 PRINT"B  FALSE"
3550 PRINT
3560 INPUT"press the letter opposite the correct answer and press ENTER";T$
3570 PRINT
3580 IF T$ = "A" THEN GOTO 3620
3590 PRINT"WRONG - READ and DATA statements DO NOT halt the program, they"
3600 PRINT"        must be used within the program."
3610 GOTO 3630
3620 PRINT"CORRECT - you certainly have a good memory."
3630 PRINT
3640 INPUT"press ENTER";T$
3650 GOSUB 6950
3660 PRINT"10 READ A1,B1"
3670 PRINT"20 PRINT A1,B1"
3680 PRINT"30 RESTORE"
3690 PRINT"40 READ C1,D1,E1"
3700 PRINT"50 PRINT C1,D1"
3710 PRINT"60 DATA 342,34"
3720 PRINT
3730 PRINT"The above program is ERROR free"
3740 PRINT
3750 PRINT"A   TRUE"
3760 PRINT"B   FALSE"
3770 PRINT
3780 INPUT"ENTER the correct answer (either A or B)";T$
3790 PRINT
3800 IF T$ = "B" GOTO 3840
3810 PRINT"WRONG - LINE 40 tried to read more DATA than was available,"
3820 PRINT"        even though the RESTORE command was used."
3830 GOTO 3850
3840 PRINT"CORRECT"
3850 PRINT
3860 INPUT"press ENTER";T$
3870 GOSUB 7100
3880 IF T$ = "B" THEN GOTO 550
3890 RETURN
3900 GOSUB 6950
3910 GOSUB 7220
3920 PRINT"There are two types of branches, and we will be studying them"
3930 PRINT"in the next two sections.  They are CONDITIONAL branches, and"
3940 PRINT"UNCONDITIONAL branches.  The above program has both kinds in it"
```

```
3950 PRINT
3960 PRINT"Line 30 is CONDITIONAL and line 40 is UNCONDITIONAL.  Can you"
3970 INPUT"see why?  Study this for a moment and then press ENTER";T$
3980 GOSUB 6950
3990 GOSUB 7220
4000 PRINT"Line 30 is CONDITIONAL because it will only GO TO line 50 if"
4010 PRINT"the CONDITION that A = 5 is satisfied.  That is, control will"
4020 PRINT"only be transferred to line 50 if A = 5."
4030 PRINT
4040 INPUT"press ENTER";T$
4050 GOSUB 6950
4060 GOSUB 7220
4070 PRINT"Line 40 is UNCONDITIONAL because it will ALWAYS GO TO line 20"
4080 PRINT"when it is executed.  There will be no choice made."
4090 PRINT"Control will go to line 20."
4100 PRINT
4110 INPUT"press ENTER";T$
4120 GOSUB 6950
4130 GOSUB 7220
4140 PRINT"Notice that A will not equal 5 until line 20 is executed 5"
4150 PRINT"times.  Therefore, until A = 5, the CONDITION in line 30 will"
4160 PRINT"NOT be met and control will NOT GOTO line 50.  Instead, it"
4170 PRINT"will go to the next line which is UNCONDITIONAL GOTO line 20."
4180 PRINT
4190 INPUT"Study this carefully, and then press ENTER";T$
4200 GOSUB 6950
4210 GOSUB 7220
4220 PRINT"The statement in line 20 has made a COUNTER out of the variable"
4230 PRINT"A.  Everytime the line is executed, A is incremented by one."
4240 PRINT"COUNTERS are very useful in BASIC and we will discuss them"
4250 PRINT"more in a future lesson.  For now, try to understand how this"
4260 PRINT"program works, and it will help you immensely in the future."
4270 PRINT
4280 INPUT"press ENTER";T$
4290 GOSUB 6950
4300 PRINT"The two types of branching are:"
4310 PRINT
4320 PRINT"A CONDITIONAL and UNCONDITIONAL"
4330 PRINT"B COUNTER and CONDITIONAL"
4340 PRINT"C COUNTER and GOTO"
4350 PRINT"D IF and GOTO"
4360 PRINT
4370 INPUT"press the letter opposite the correct answer and press ENTER";T$
4380 PRINT
4390 IF T$ = "A" THEN GOTO 4420
4400 PRINT"WRONG - the correct answer is A (CONDITIONAL and UNCONDITIONAL)"
```

```
4410 GOTO 4470
4420 PRINT"CORRECT"
4430 PRINT
4440 INPUT"press ENTER";I$
4450 GOSUB 6950
4460 PRINT"10 A = 0"
4470 PRINT"20 IF A = 3 GOTO 50"
4480 PRINT"30 A = A+1"
4490 PRINT"40 GOTO 20"
4500 PRINT"50 PRINT A"
4510 PRINT"60 END"
4520 PRINT"RUN"
4530 PRINT
4540 PRINT"what would be the output of this program"
4550 PRINT
4560 PRINT"A  1"
4570 PRINT"B  2"
4580 PRINT"C  no output would come from this program"
4590 PRINT"D  3"
4600 PRINT
4610 INPUT"press the letter opposite the correct answer and press ENTER";I$
4620 PRINT
4630 IF I$ = "D" THEN GOTO 4660
4640 PRINT"WRONG - the correct answer is D (3)"
4650 GOTO 4670
4660 PRINT"CORRECT"
4670 PRINT
4680 INPUT" press ENTER";I$
4690 GOSUB 7100
4700 IF Y$ = "Y" GOTO 2900
4710 RETURN
4720 GOSUB 6950
4730 PRINT"                    IF Statements"
4740 PRINT
4750 PRINT"IF Statements are decision makers in BASIC.  They test to see"
4760 PRINT"IF a condition is met. If it is, THEN they execute the commands"
4770 PRINT"that follow them on the same line."
4780 PRINT
4790 PRINT"The IF statement causes the program to make comparisons between"
4800 PRINT" values.  It is one of the most powerful commands in the BASIC"
4810 PRINT"language.  You have already seen how it can be used in the"
4820 PRINT"previous section.  In this section, we will explain it in a"
4830 PRINT" little more detail."
4840 PRINT
4850 INPUT" press ENTER";I$
4860 GOSUB 695
```

```
4870 GOSUB 7250
4880 PRINT"This example shows three variations of the IF statement."
4890 PRINT"If we RUN this program, and ENTER a '0' when prompted by line"
4900 PRINT"10, then the CONDITIONAL statement in line 20 will be satisfied"
4910 PRINT"and the program will stop."
4920 PRINT
4930 INPUT"press ENTER";T$
4940 GOSUB 6950
4950 GOSUB 7250
4960 PRINT"If we ENTER a 12 when prompted by line 10, the CONDITIONAL"
4970 PRINT"statement in line 20 will not be satisfied, nothing will happen"
4980 PRINT"until control passes to line 30.  At that time, the check for"
4990 PRINT"N GREATER THAN 10 will be met, and the message will be printed."
5000 PRINT"No other condition will be met until control gets to line 50."
5010 PRINT"The UNCONDITIONAL GOTO on line 50 will send control back to the"
5020 PRINT"beginning of the program."
5025 PRINT
5030 INPUT"press ENTER to continue";T$
5040 GOSUB 6950
5050 GOSUB 7250
5060 PRINT"If we ENTER a -12 when prompted by line 10, we will satisfy the"
5070 PRINT CONDITIONAL statement in line 40, the message will be printed"
5080 PRINT"and control will eventually get back to line 10."
5090 PRINT
5100 INPUT"press ENTER";T$
5110 GOSUB 6950
5120 PRINT"                    IF Statement (cont)"
5130 PRINT
5140 PRINT"The IF statement can also be used to compare two expressions"
5150 PRINT"such as:"
5160 PRINT
5170 PRINT"20 IF (10*2-7    (3*10+6) THEN GOTO 100"
5180 PRINT
5190 PRINT"Also, variable assignment can be done in an IF statement:"
5200 PRINT
5210 PRINT"20 IF (20*2-7    3*10) THEN A=1"
5220 PRINT
5230 PRINT
5240 INPUT"press ENTER";T$
5250 GOSUB 6950
5260 PRINT"There is another word that can be added to the IF statement"
5270 PRINT"to make it more powerful.  It is the ELSE word."
5280 PRINT
5290 PRINT"20  IF A = 1 THEN GOTO 10 ELSE GOTO 100"
5300 PRINT
5310 PRINT"In this line, if the variable A equals 1 then control transfers"
```

```
5340 PRINT"to line 10. If it does NOT equal 1 then control transfers to "
5350 PRINT"200.  In this case, something ALWAYS happens at line 220"
5360 PRINT"because of the ELSE statement."
5370 PRINT
5380 PRINT"220 IF A = 1 THEN GOTO 10 ELSE IF A = 2 GOTO 30"
5390 PRINT
5400 PRINT"In this case, if A = 1   or  A = 2 then something will happen"
5410 PRINT"in line 220, if none of the CONDITIONS are met, then the line"
5420 PRINT"will not be executed."
5425 PRINT
5430 INPUT"press ENTER";T$
5440 GOSUB 6950
5450 PRINT"                    IF Statement (cont)"
5460 PRINT
5470 PRINT"IF (true/false expression) THEN (action) ELSE (action)"
5480 PRINT
5490 PRINT"The IF statement instructs the computer to test the following"
5500 PRINT"logical or relational expression.  If the expression is TRUE"
5510 PRINT"then control will proceed to the action line after the THEN"
5520 PRINT"word.  If the expression is not true, then control will proceed"
5530 PRINT"to the ELSE action."
5540 PRINT
5550 INPUT"press ENTER";T$
5560 GOSUB 6950
5570 PRINT"What is the output of the following program?"
5580 PRINT
5590 PRINT"10 A = 255"
5600 PRINT"20 IF A   190 THEN PRINT "CHR$(34)"TOO WEAK"CHR$(34)
5610 PRINT"30 IF A > 254 THEN PRINT "CHR$(34)"TOO STRONG"CHR$(34)
5620 PRINT"40 IF A = 255 THEN PRINT "CHR$(34)"A = 255"CHR$(34)
5630 PRINT
5640 PRINT"A  TOO WEAK"
5650 PRINT"   A = 255"
5660 PRINT"B  TOO STRONG"
5670 PRINT"   END"
5680 PRINT"C  TOO STRONG"
5690 PRINT"   A = 255"
5700 PRINT"D  A = 255"
5710 PRINT
5720 INPUT"ENTER the letter opposite the correct answer";T$
5730 PRINT
5740 IF T$ = "C" GOTO 5780
5750 PRINT"WRONG - the correct answer is C (TOO STRONG)"
5760 PRINT"                                (A = 255)"
5770 GOTO 5790
5780 PRINT"CORRECT - We need your mind in the budget office!"
```

```
5790 PRINT
5800 INPUT"press ENTER";T$
5810 GOSUB 6950
5820 PRINT"The IF statement is a CONDITIONAL statement."
5830 PRINT
5840 PRINT"Is the above sentence TRUE or FALSE?"
5850 PRINT
5860 PRINT"A  TRUE"
5870 PRINT"B  FALSE"
5880 PRINT
5890 INPUT"choose the letter opposite the correct answer and press ENTER";T$
5900 PRINT
5910 IF T$ = "A" GOTO 5940
5920 PRINT"WRONG - the IF statement IS a CONDITIONAL statement."
5930 GOTO 5950
5940 PRINT"CORRECT"
5950 PRINT
5960 INPUT"press ENTER";T$
5970 GOSUB 7100
5980 IF T$ = "B" GOTO 4720
5990 RETURN
6000 GOSUB 6950
6010 PRINT"                    GOTO Statements"
6020 PRINT
6030 PRINT"Conditional branches are written in programs with IF THEN ELSE"
6040 PRINT"statements.  Unconditional branches are written with GOTO"
6050 PRINT"statements."
6060 PRINT"As we saw earlier, GOTO directs control of a program to another"
6070 PRINT"line.  For example:"
6080 GOSUB 7470
6090 PRINT
6100 INPUT "press ENTER";T$
6110 GOSUB 6950
6120 GOSUB 7470
6130 PRINT
6140 PRINT"The GOTO statement in line 40, when executed, sends control to"
6150 PRINT"the beginning of the program."
6160 PRINT
6170 INPUT"press ENTER";T$
6180 GOSUB 6950
6190 PRINT"                    GOTO (cont)"
6200 PRINT
6210 PRINT"You can make the GOTO statement a MULTI-way branching statement"
6220 PRINT"by modifying it slightly.  For example:"
6230 PRINT
6240 GOSUB 7500
```

```
6250 PRINT
6260 INPUT"press ENTER";T$
6270 GOSUB 6950
6280 GOSUB 7580
6290 PRINT
6300 PRINT"When line 20 is executed, the value of N is used to count over"
6310 PRINT"'N' elements passed the GOTO word.  Control branches to the"
6320 PRINT"line number indicated by this 'Nth' element.  If there is no"
6330 PRINT"element that corresponds to the value of N, then control passes"
6340 INPUT"to the next available line.  Press ENTER when ready";T$
6350 GOSUB 6950
6360 GOSUB 7580
6370 PRINT
6380 PRINT"The value of N MUST be greater than 0 and less than 255.  If it"
6390 PRINT"is not, BASIC will print an error.  If N is 1, 2, or 3 then the"
6400 PRINT"program will print the appropriate message and stop."
6410 PRINT
6420 INPUT"press ENTER";T$
6430 GOSUB 6950
6440 PRINT"Which lines (beyond 20) are executed if you ENTER a 10 here?"
6450 PRINT
6460 GOSUB 7580
6470 PRINT
6480 PRINT"A  30 and 40"
6490 PRINT"B  50 and 60"
6500 PRINT"C  70 and 80"
6510 PRINT
6520 INPUT"press the letter opposite the correct answer then pres ENTER";T$
6530 PRINT
6540 IF T$ = "A" GOTO 6570
6550 PRINT"WRONG - the correct answer is A (30 and 40)"
6555 PRINT
6560 GOTO 6590
6570 PRINT"CORRECT"
6580 PRINT
6590 INPUT"press ENTER";T$
6640 GOSUB 6950
6810 PRINT"You are now done with this lesson.  When you hit ENTER, you"
6820 PRINT"will be returned to the MENU where you may review sections"
6830 PRINT"or take the TEST."
6840 PRINT
6850 PRINT"Remember, after you are done here, practice some of the"
6860 PRINT"things you have learned.  And keep a BASIC manual by your side."
6870 PRINT"This program will show you the fundamentals, you have to teach"
6880 PRINT"yourself how to be good at BASIC.  That means you must PRACTICE"
6890 PRINT
```

```
6900 INPUT"press ENTER to go to the MENU";I$
6910 RUN
6920 REM **
6930 REM ** This subroutine clears the screen on any terminal
6940 REM **
6950 FOR X = 1 TO 24
6960    PRINT
6970 NEXT X
6980 RETURN
6990 PRINT"                    LESSON 2B"
7000 PRINT
7010 PRINT"This is the second part of a two part lesson"
7020 PRINT"It is divided into the following sections."
7030 PRINT
7040 PRINT"1)  READ, DATA and RESTORE    3)  IF Statements"
7050 PRINT"2)  Branching Introduction    4)  GOTO Statements & Summary"
7060 PRINT"                    5)  TEST "
7070 PRINT
7080 PRINT
7090 RETURN
7100 GOSUB 6950
7110 PRINT"Which do you want to do?"
7120 PRINT
7130 PRINT"A  Continue on"
7140 PRINT"B  Review this lesson again"
7150 PRINT
7160 INPUT"press the letter opposite your choice and press ENTER";I$
7170 IF I$ <> "A" AND I$ <> "B" GOTO 7160
7180 RETURN
7190 REM
7200 REM This subroutine prints the Branching section example
7210 REM
7220 PRINT"                    Branching Introduction"
7230 PRINT
7240 PRINT"10 A = 0"
7250 PRINT"20 A = A+1"
7260 PRINT"30 IF A = 5 THEN GOTO 50"
7270 PRINT"40 GOTO 20"
7280 PRINT"50 PRINT A"
7290 PRINT"60 END"
7300 PRINT
7310 RETURN
7320 REM
7330 REM This subroutine is for the IF statement examples
7340 REM
7350 PRINT"                    IF Statements (cont)"
```

```
7360 PRINT
7370 PRINT"10 INPUT "CHR$(34)"ENTER a number between 1 and 10 (0 to quit)"CHR$(34)";N"
7380 PRINT"20 IF N = 0 THEN STOP"
7390 PRINT"30 IF N > 10 THEN PRINT "CHR$(34)"ERROR - you entered an invalid number"CHR$(34)
7400 PRINT"40 IF N < 0 PRINT "CHR$(34)"ERROR - you entered an invalid number"CHR$(34)
7410 PRINT"50 GOTO 10"
7420 PRINT
7430 RETURN
7440 REM
7450 REM this is subroutine for GOTO example
7460 REM
7470 PRINT
7480 PRINT"10 INPUT "CHR$(34)"ENTER a number between 1 and 10"CHR$(34)";N"
7490 PRINT"20 IF N = 8 THEN GOTO 50"
7500 PRINT"30 IF N <> 8 THEN PRINT "CHR$(34)"GUESS AGAIN"CHR$(34)
7510 PRINT"40 GOTO 10"
7520 PRINT"50 PRINT "CHR$(34)"YOU GUESSED IT"
7530 PRINT"60 END"
7540 RETURN
7550 REM
7560 REM this is the subroutine example for ON GOTO
7570 REM
7580 PRINT"10 INPUT "CHR$(34)"ENTER a number between 1 and 3"CHR$(34)";N"
7590 PRINT"20 ON N GOTO 30,50,70
7600 PRINT"30 PRINT "CHR$(34)"YOU ENTERED A ONE (or an illegal number)"CHR$(34)
7610 PRINT"40 STOP"
7620 PRINT"50 PRINT "CHR$(34)"YOU ENTERED A TWO"CHR$(34)
7630 PRINT"60 STOP"
7640 PRINT"70 PRINT "CHR$(34)"YOU ENTERED A THREE"CHR$(34)
7650 PRINT"80 STOP"
7660 RETURN
7670 REM
7680 RUN"TEST2"
7690 RUN"MENU"
7700 RUN"LESSON2"
7710 END
```

***** Listing of Program 'TEST2' ***** 07/10/83 - 00:48:52

```
10 REM **
20 REM ** LESSON:  TEST2                VERSION: 1 AUG 83
30 REM ** AUTHOR:  CAPT DAN CREAGAN
40 REM **          AIR FORCE INSTITUTE OF TECHNOLOGY
50 REM **
60 REM ** VARIABLES:
70 REM **            N$(X) = NAMES ARRAY, USED TO READ IN SEQ-
80 REM **                    UENTIAL NAMES, AND TO WRITE OUT
90 REM **                    UPDATE NAMES.
100 REM **            S(X)  = SCORES ARRAY - USED TO READ AND
110 REM **                    WRITE SCORES
120 REM **            Q(X)  = ARRAY TO KEEP TRACK OF NUMBER OF
130 REM **                    CORRECT ANSWERS.  IF AN ARRAY
140 REM **                    ELEMENT EQUALS 1, THE ANSWER WAS
150 REM **                    CORRECT
160 REM **
170 CLEAR 3000
180 GOSUB 3280
190 DIM N$(1000)
200 DIM Q(10)
210 DIM S(1000)
220 PRINT"                    FINAL TEST (lesson 2)"
230 PRINT
240 PRINT"This test consists of 10 questions, you must get 70 percent"
250 PRINT"of them correct to pass.  (that's 7 right out of the 10 ques-"
260 PRINT"tions).  Use only capital letters in your answers, don't"
270 PRINT"include extra spaces or letters.  GOOD LUCK"
280 PRINT
290 INPUT"press ENTER to continue";T$
300 GOSUB 3280
310 PRINT"Which of the following is a legal filename?"
320 PRINT
330 PRINT"A  50DIXIE.CMD (in CPM or Cromemco)"
340 PRINT"B  TRIUMPH/650 (in TRS-80)"
350 PRINT"C  THEWAYOF.921 (in CPM or Cromemco)"
360 PRINT"D  SCORE?/DAT (in TRS-90)"
370 PRINT
380 INPUT"ENTER the letter opposite the correct answer";T$
390 PRINT
400 IF T$ = "C" THEN GOTO 470
410 PRINT"WRONG - the correct answer is C"
420 PRINT"        Answers A and B filenames or extensions start with"
430 PRINT"        numbers instead of alphabet characters. Answer D"
440 PRINT"        has a non alpha-numeric character in it."
450 PRINT"        See part 1, filenames, in lesson 2."
460 GOTO 490
```

```
470 PRINT"CORRECT"
480 Q(1) = 1
490 PRINT
500 INPUT"press ENTER";T$
510 GOSUB 3280
520 PRINT"Which of the following commands will load a file called 'LOVE'?"
530 PRINT
540 PRINT"A  CREATE "CHR$(34)"LOVE"CHR$(34)
550 PRINT"B  RUN "CHR$(34)"LOVE"CHR$(34)
560 PRINT"C  LOAD "CHR$(34)"LOVE.BAS2"CHR$(34)
570 PRINT"D  SAVE "CHR$(34)"LOVE"CHR$(34)
580 PRINT
590 INPUT "ENTER the correct answer";T$
600 PRINT
610 IF T$ = "B" THEN GOTO 680
620 PRINT"WRONG - the correct answer is B"
630 PRINT"        In answer A, CREATE is not a BASIC word, in C"
640 PRINT"        an incorrect extension was used (no extension was"
650 PRINT"        needed), in D the program would be saved, not loaded."
660 PRINT"        See part 1, SAVE, LOAD, RUN of lesson 2."
670 GOTO 700
680 PRINT"CORRECT"
690 Q(2) = 1
700 PRINT
710 INPUT"press ENTER";T$
720 GOSUB 3280
730 PRINT"REMark statements are similar to PRINT statements, except that"
740 PRINT"the computer takes less time to print them."
750 PRINT
760 PRINT"A  TRUE"
770 PRINT"B  FALSE"
780 PRINT
790 INPUT"ENTER the letter opposite the correct answer";T$
800 PRINT
810 IF T$ = "B" GOTO 860
820 PRINT"WRONG - the correct answer is B"
830 PRINT"        REMark statements are not output to the screen, they"
840 PRINT"        are only used for programmer information. See part 1"
850 GOTO 880
860 PRINT"CORRECT"
870 Q(3) = 1
880 PRINT
890 INPUT"press ENTER";T$
900 GOSUB 3280
910 PRINT"Which of the following examples is INVALID?"
920 PRINT
```

```
930 PRINT"A  INPUT T$"
940 PRINT"B  INPUT"CHR$(34)"ENTER YOUR NAME"CHR$(34)";N"
950 PRINT"C  INPUT N"
960 PRINT"D  INPUT"CHR$(34)"ENTER YOUR AGE"CHR$(34)";N"
970 PRINT
980 INPUT"ENTER the letter opposite the correct answer";T$
990 PRINT
1000 IF T$ = "B" THEN GOTO 1050
1010 PRINT"WRONG - statement B is the bad one because it tries"
1020 PRINT"        to load a numeric variable with string data."
1030 PRINT"        See Part 1."
1040 GOTO 1070
1050 PRINT"CORRECT"
1060 Q(4) = 1
1070 PRINT
1080 INPUT"press ENTER";T$
1090 GOSUB 3280
1100 PRINT"What command will let you put your program into permanent"
1110 PRINT"storage so that you can recall it later?"
1120 PRINT
1130 PRINT"A  SAVE"
1140 PRINT"B  RUN"
1150 PRINT"C  LOAD"
1160 PRINT"D  STORE"
1170 PRINT
1180 INPUT"ENTER the letter opposite the correct answer";T$
1190 PRINT
1200 IF T$ = "A" THEN GOTO 1260
1210 PRINT"WRONG - the correct answer is A"
1220 PRINT"        RUN causes program execution, LOAD loads the "
1230 PRINT"        program from disk and STORE is not a BASIC word."
1240 PRINT"        See part 1."
1250 GOTO 1280
1260 PRINT"CORRECT"
1270 Q(5) = 1
1280 PRINT
1290 INPUT"press ENTER";T$
1300 GOSUB 3280
1310 PRINT"Which of the following statements is legal?"
1320 PRINT
1330 PRINT"A  READ A$ B% C1"
1340 PRINT"B  DATA A B C"
1350 PRINT"C  READ 22,33.44"
1360 PRINT"D  DATA "CHR$(34)"QUIT"CHR$(34)","CHR$(34)"FIRE"CHR$(34)
1370 PRINT
1380 INPUT"ENTER the letter opposite the correct answer";T$
```

```
1390 PRINT
1400 IF T$ = "D" GOTO 1470
1410 PRINT"WRONG - the correct answer is D"
1420 PRINT"        Answers A and B don't have commas between variables,"
1430 PRINT"        and answer C tries to use constants instead of"
1440 PRINT"        variables for the READ."
1450 PRINT"        See part 2, READ and DATA."
1460 GOTO 1490
1470 PRINT"CORRECT"
1480 Q(6) = 1
1490 PRINT
1500 INPUT"press ENTER";T$
1510 GOSUB 3280
1520 PRINT"READ and DATA statements halt the program so the operator can"
1530 PRINT"insert correct answers."
1540 PRINT
1550 PRINT"A  TRUE"
1560 PRINT"B  FALSE"
1570 PRINT
1580 INPUT"ENTER the letter opposite the correct answer";T$
1590 PRINT
1600 IF T$ = "B" GOTO 1660
1610 PRINT"WRONG - the correct answer is B"
1620 PRINT"        READ and DATA are used to load variables WITHOUT"
1630 PRINT"        stopping the program."
1640 PRINT"        See part 2, READ and DATA."
1650 GOTO 1680
1660 PRINT"CORRECT"
1670 Q(7) = 1
1680 PRINT
1690 INPUT"press ENTER";T$
1700 GOSUB 3280
1710 PRINT"The IF statement is a CONDITIONAL BRANCHING statement."
1720 PRINT
1730 PRINT"A  TRUE"
1740 PRINT"B  FALSE"
1750 PRINT
1760 INPUT"ENTER the letter opposite the correct answer";T$
1770 PRINT
1780 IF T$ = "A" GOTO 1820
1790 PRINT"WRONG - the correct answer is A"
1800 PRINT"        See Part 2, IF and GOTO."
1810 GOTO 1840
1820 PRINT"CORRECT"
1830 Q(8) = 1
1840 PRINT
```

```
1850 INPUT"press ENTER";T$
1860 GOSUB 3280
1870 PRINT"The following program will NOT have an output - TRUE or FALSE?"
1880 PRINT
1890 PRINT"10 N = 5"
1900 PRINT"20 ON N GOTO 40,60,80"
1910 PRINT"30 STOP"
1920 PRINT"40 PRINT N"
1930 PRINT"50 STOP"
1940 PRINT"60 PRINT N"
1950 PRINT"70 STOP"
1960 PRINT"80 PRINT N"
1970 PRINT"90 STOP"
1980 PRINT
1990 PRINT"A  TRUE"
2000 PRINT"B  FALSE"
2010 INPUT"ENTER the letter opposite the correct answer";T$
2020 PRINT
2030 IF T$ = "A" GOTO 2090
2040 PRINT"WRONG - the correct answer is A"
2050 PRINT"        Line 10 sets N to 5, line 20  only has 3 places"
2060 PRINT"        to go to, so it defaults to the line under it."
2070 PRINT"        That line is a STOP statement. See part 2, GOTO."
2080 GOTO 2110
2090 PRINT"CORRECT"
2100 Q(9) = 1
2110 PRINT
2120 INPUT"press ENTER";T$
2130 GOSUB 3280
2140 PRINT"The following program will have an output - TRUE or FALSE?"
2150 PRINT
2160 PRINT"10 N = 3"
2170 PRINT"20 IF N = 3 THEN GOTO 40"
2180 PRINT"30 PRINT N"
2190 PRINT"40 N = 4"
2200 PRINT"50 END"
2210 PRINT
2220 PRINT
2230 PRINT"A  TRUE"
2240 PRINT"B  FALSE"
2250 PRINT
2260 INPUT"ENTER the letter opposite the correct answer";T$
2270 PRINT
2280 IF T$ = "B" GOTO 2340
2290 PRINT"WRONG - the correct answer is B"
2300 PRINT"        Line 10 sets N to 3, line 20 causes the program to"
```

```
2310 PRINT"       go to line 40, then 50."
2320 PRINT"       See part 2, IF."
2330 GOTO 2360
2340 PRINT"CORRECT"
2350 Q(10) = 1
2360 PRINT
2370 INPUT"press ENTER";T$
2380 GOSUB 3280
2390 FOR X = 1 TO 10
2400    Y = Y+Q(X)
2410 NEXT X
2420 PRINT"You have finished the test, out of 10 possible correct answers"
2430 PRINT"you scored "Y"."
2440 PRINT
2450 IF Y > 6 THEN  PRINT"YOU HAVE PASSED"
2460 GOSUB 3110
2470 IF Y > 6 THEN GOTO 2550
2480 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
2490 PRINT
2500 PRINT"YOU SHOULD RETAKE LESSON 2!"
2510 PRINT
2520 PRINT"You will be returned to the Menu."
2530 PRINT
2540 GOTO 3320
2550 PRINT
2560 PRINT"Do you want your score recorded on a permanent file?"
2570 PRINT
2580 PRINT"A  YES"
2590 PRINT"B  NO"
2600 PRINT
2610 INPUT"Which";T$
2620 IF T$ = "B" THEN GOTO 2920
2630 GOSUB 3280
2640 PRINT"To record your score, we must open a file and put your name"
2650 PRINT"in it.  Therefore, surprisingly, we need your name.  If your"
2660 PRINT"name is not unique among the students likely to take this test,"
2670 PRINT"please contact your test monitor for an identifying word that"
2680 PRINT"will make you unique.  Then enter that word below."
2690 PRINT
2700 PRINT"If you have already entered a score previously, be sure to"
2710 PRINT"enter the same name you used before.  (use all capitals!"
2720 PRINT
2730 INPUT"ENTER your word or name now";T$
2740 OPEN"I",1,"SCORE2"
2750 I = 0
2760 IF EOF(1) THEN GOTO 2820
```

```
2770 X = X+1
2780 INPUT#1,N$(X)
2790 INPUT#1,S(X)
2800 IF N$(X) = T$ THEN GOTO 2970
2810 GOTO 2760
2820 CLOSE
2830 X = X+1
2840 N$(X) = T$
2850 S(X) = Y
2860 OPEN"O",1,"SCORE2"
2870 FOR W = 1 TO X
2880   PRINT#1,N$(W)
2890   PRINT#1,S(W)
2900 NEXT W
2910 PRINT
2920 PRINT"You are now qualified to go to LESSON 3, however, you will"
2930 PRINT"first get a homework assignment ! The homework is in a "
2940 PRINT"program module. If you don't want the assignment, you may"
2950 PRINT"go to the MENU instead."
2960 GOTO 3250
2970 S(X) = Y
2980 IF EOF(1) THEN CLOSE:GOTO 3020
2990 X = X+1
3000 INPUT#1, N$(X), S(X)
3010 GOTO 2980
3020 OPEN"O",1,"SCORE2"
3030    FOR W = 1 TO X
3040        PRINT#1,N$(W)
3050        PRINT#1,S(W)
3060    NEXT W
3070 PRINT
3080 PRINT"You may now take LESSON 3.  You will be returned to the MENU"
3090 PRINT"from where you may go to LESSON 3 or quit."
3100 GOTO 3320
3110 IF Y=10 THEN RETURN
3120 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
3130 PRINT
3140 IF Q(1) = 0 THEN PRINT"    part 1, Filenames"
3150 IF Q(2) = 0 OR Q(5) = 0  THEN PRINT"    part 1, SAVE, LOAD, RUN"
3160 IF Q(3) = 0 THEN PRINT"    part 1, REMarks"
3170 IF Q(4) = 0 THEN PRINT"    part 1, INPUT Statements"
3180 IF Q(6) = 0 OR Q(7) = 0 THEN PRINT"    part 2, READ and DATA Statements"
3190 IF Q(8) = 0 OR Q(9) = 0 THEN PRINT"    part 2, IF Statements"
3200 IF Q(10) = 0 THEN PRINT"    part 2, GOTO Statements"
3210 PRINT
3220 INPUT"press ENTER";T$
```

```
3230 GOSUB 3280
3240 RETURN
3250 REM **
3260 REM ** this subroutine clears the screen*
3270 REM **
3280 FOR X = 1 TO 24
3290     PRINT
3300 NEXT X
3310 RETURN
3320 PRINT
3330 INPUT"press ENTER to return to the MENU";T$
3340 RUN "MENU"
3350 PRINT
3360 INPUT"Do you want the assignment (Y/N)";T$
3370 IF T$ = "N" THEN GOTO 3340
3380 RUN "HW2"
```

```
 10 REM ** THIS PROGRAM STARTED ON 1 MAY 1983
 20 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
 30 REM ** TITLE: LESSON 3
 40 REM **
 50 REM **
 60 REM **
 70 REM **
 80 GOSUB 5590
 90 PRINT"LESSON:  BASIC 3            VERSION: 1 AUGUST 83
100 PRINT
110 PRINT"TIME REQUIRED TO COMPLETE LESSON:  Less than one hour"
120 PRINT
130 PRINT
140 PRINT"AUTHOR:  Capt Danny J. Creagan"
150 PRINT"         Air Force Institute of Technology"
160 PRINT
170 PRINT"OBJECTIVE:  To teach the student how to use LOOPS and ARRAYS."
180 PRINT
190 PRINT
200 PRINT
210 PRINT
220 PRINT
230 PRINT
240 INPUT"press the ENTER key to continue";T$
250 GOSUB 5590
260 GOSUB 5660
270 PRINT"A  I'm taking this part in its entirety."
280 PRINT"B  I wish to review selected areas."
290 PRINT"C  I want to go to the second part."
300 PRINT"D  I want to return to the Menu."
310 PRINT
320 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
330 IF T$ = "D" GOTO 6370
340 IF T$ = "C" GOTO 6380
350 IF T$ = "B" GOTO 460
360 IF T$ <>"A" GOTO 270
370 GOSUB 570
380 GOSUB 1090
390 GOSUB 1450
400 GOSUB 2240
410 GOSUB 3430
420 PRINT
430 PRINT
440 PRINT"Going to second half of lesson 3 - Wait one moment"
450 GOTO 6380
460 GOSUB 5590
```

```
470 GOSUB 5660
480 PRINT
490 PRINT"Please type in the number beside the area you wish"
500 PRINT"to review (1 through 5) and then press ENTER - press 0 and"
510 PRINT"press ENTER to return to the Menu."
520 PRINT
530 INPUT"What is your choice";N
540 IF N = 0 GOTO 6370
550 ON N GOSUB 570  ,1090 ,1450 ,2240 ,3430
560 GOTO 460
570 GOSUB 5590
580 PRINT"                    Introduction & KILL Statement"
590 PRINT
600 PRINT"By now you should have saved a few small programs to disk, and"
610 PRINT"you should have practiced all the commands we have discussed to"
620 PRINT"date.  If you have wondered how to get rid of a file that"
630 PRINT"you were done with, or have mistakenly saved and didn't need"
640 PRINT"one of your programs, the next frame will be of service to"
650 PRINT"you.  It describes the KILL statement.  We have purposely"
660 PRINT"put the KILL statement in the third lesson because it is a"
670 PRINT"dangerous command that can eliminate valuable and irreplacable"
680 PRINT"data or programs if used incorrectly.  By now you should feel"
690 PRINT"comfortable with some of the commands and you shouldn't make"
700 PRINT"the mistake of KILLing someone else's files, or your own by"
710 PRINT"accident.  Remember, KILL does just what it says, it KILLs"
720 PRINT"files!"
725 PRINT
730 INPUT"press ENTER to learn about the KILL command";T$
740 GOSUB 5590
750 PRINT"                    Introduction & KILL (cont)"
760 PRINT
770 PRINT"The format for the  KILL command is:"
780 PRINT
790 PRINT"KILL "CHR$(34)"filename.ext"CHR$(34)
800 PRINT
810 PRINT"You must enclose the filename in quotation marks."
812 PRINT"The extension is only needed when the original file had one."
820 PRINT
830 PRINT"Once this command is entered, the file will be removed (unless"
840 PRINT"protected by passwords - you can learn about passwords from"
950 PRINT"your system  manual).  If the file is removed, there is very"
860 PRINT"little chance of recovering any of your data.  In some cases"
870 PRINT"an advanced programmer can retrieve data from a killed file"
880 PRINT"but the process is difficult and often fails.  BE CAREFUL!"
890 PRINT
900 INPUT"press ENTER";T$
```

```
910 GOSUB 5590
920 PRINT"Which of the following statements is legal?"
930 PRINT
940 PRINT"A  KILL STUPID"
950 PRINT"B  KILL "CHR$(34)"STUPID.123"CHR$(34)
960 PRINT"C  KILL "CHR$(34)"STUPID.BAS"CHR$(34)
970 PRINT
980 INPUT"press the letter opposite the correct answer and press ENTER";T$
990 PRINT
1000 IF T$ = "C" GOTO 1030
1010 PRINT"WRONG - the correct answer is C"
1020 GOTO 1040
1030 PRINT"CORRECT"
1040 PRINT
1050 INPUT"press ENTER";T$
1060 GOSUB 5800
1070 IF T$ = "B" GOTO 570
1080 RETURN
1090 GOSUB 5590
1100 PRINT"                    LOOPs (Intro)"
1110 PRINT
1120 PRINT"One of the most exciting aspects of computer programs is their"
1130 PRINT"ability to patiently check and recheck data, and to tirelessly"
1140 PRINT"calculate figures and columns of numbers.  (the exciting part"
1150 PRINT"is that YOU don't have to spend hours doing drudgery when the"
1160 PRINT"computer can spend minutes or seconds doing the same job)"
1170 PRINT"One of the tasks of the programmer is to efficiently use his"
1180 PRINT"computer memory to program the time consuming tasks."
1190 PRINT
1200 INPUT"press ENTER";T$
1210 GOSUB 5590
1220 PRINT"Suppose you wanted to display 3 columns of figures, the left"
1230 PRINT"column would be integers from 1 to 8, the middle column"
1240 PRINT"would be the square of the figure in the adjacent first column,"
1250 PRINT"and the third column would be the square of the figure in the"
1260 PRINT"adjacent middle column.  It would look like this:"
1270 PRINT
1280 FOR X = 1 TO 8
1290   PRINT INT(X),INT(X^2),INT(X^2*X^2)
1300 NEXT X
1310 PRINT
1320 INPUT"press ENTER";T$
1330 GOSUB 5590
1340 FOR X = 1 TO 8
1350   PRINT INT(X),INT(X^2),INT(X^2*X^2)
1360 NEXT X
```

```
1370 PRINT
1380 PRINT"If you were to write separate print statements for these"
1390 PRINT"calculations, you would have to write at least 8 lines of code."
1400 PRINT"Using LOOPs, you can calculate this column of figures, and"
1410 PRINT"display it, in three lines of code!"
1420 PRINT
1430 INPUT"In the next sections we will learn this and more...press ENTER";T$
1440 RETURN
1450 GOSUB 5590
1460 PRINT"                         COUNTER variables"
1470 PRINT
1480 PRINT"The first step in learning about loops is to understand COUNTER"
1490 PRINT"variables.  A COUNTER is a variable that is used to keep track"
1500 PRINT"of the number of times a program executes a line.  It is "
1510 PRINT"often used in conjunction with an IF THEN statement.  For"
1520 PRINT"example:"
1530 GOSUB 6050
1540 PRINT"The value of Z after RUNning this program is 10."
1550 PRINT
1560 INPUT"press ENTER";T$
1570 GOSUB 5590
1580 PRINT"                       COUNTERs (cont)"
1590 PRINT
1600 GOSUB 6050
1610 PRINT"The COUNTER variable is Z.  It is INCREMENTED every time line"
1620 PRINT"40 sends control to it.  When the CONDITIONAL statement in"
1630 PRINT"line 30 discovers that Z is equal to 10, it executes the STOP."
1640 PRINT"At the end of the program, Z is equal to 10."
1650 PRINT
1660 INPUT"press ENTER";T$
1670 GOSUB 5590
1680 GOSUB 6050
1690 PRINT"Do you see what would happen if we didn't include the IF"
1700 PRINT"statement?  The program would look like this:"
1710 PRINT
1720 PRINT"10 Z=0"
1730 PRINT"20 Z=Z+1"
1740 PRINT"30 GOTO 20"
1750 PRINT
1760 PRINT"The program would never stop.  It would be caught in an ENDLESS"
1770 PRINT"LOOP.  We have many terms for this common mistake.  Some"
1780 PRINT"examples: LOCKED UP, HUNG, and *** DUMMY, you did it again! ***"
1790 PRINT
1800 INPUT"press ENTER";T$
1810 GOSUB 5590
1820 PRINT"                       COUNTERs (cont)"
```

```
1820 PRINT
1830 PRINT"If your program ever gets hung up, you may have to type "
1840 PRINT"<CONTROL> C (if CPM or Cromemco), or <BREAK> (if TRS-80)"
1850 PRINT"In some extreme cases, you may have to reset the system."
1860 PRINT"On the TRS-80, the RESET button is the red recessed button on"
1870 PRINT"the upper right of your keyboard (left rear if using a MODEL I)"
1880 PRINT"CAUTION, if you are using this program on a multi-user system,"
1890 PRINT"such as Cromemco System II, DO NOT RESET the system, let the"
1900 PRINT"operator help you fix the problem.  If you are using a single"
1910 PRINT"user system, such as a TRS-80, then RESET will work as an exit"
1920 PRINT"from the locked up program (but try <BREAK> first).  If you use"
1930 PRINT"either RESET or <CONTROL> C, the system will take you to the"
1940 PRINT"COMMAND mode, and you will have to re-initialize BASIC."
1950 PRINT
1960 INPUT"press ENTER";I$
1970 GOSUB 5590
1980 PRINT"What is the COUNTER variable in this program?"
1990 PRINT
2000 PRINT"10 Z=0"
2010 PRINT"20 X=0"
2020 PRINT"30 X=X+1"
2030 PRINT"40 IF Z = X THEN Z = Z"
2040 PRINT"50 IF X = 5 THEN STOP"
2050 PRINT"60 PRINT Z"
2060 PRINT"70 GOTO  30"
2070 PRINT"80 END"
2080 PRINT
2090 PRINT"A  Z is the COUNTER"
2100 PRINT"B  X is the COUNTER"
2110 PRINT"C  BOTH are COUNTERs, but X will control the program"
2120 PRINT
2130 INPUT"press the letter opposite the correct answer and press ENTER";I$
2140 PRINT
2150 IF I$ = "B" THEN GOTO 2180
2160 PRINT"WRONG - the correct answer is B"
2170 GOTO 2190
2180 PRINT"CORRECT - that shows good understanding of the principle!"
2190 PRINT
2200 INPUT"press ENTER";I$
2210 GOSUB 5590
2220 IF I$ = "B" GOTO 1450
2230 RETURN
2240 GOSUB 5590
2250 PRINT"                    FOR - NEXT Statements"
2260 PRINT
2270 PRINT"In the previous discussion of counters, we showed you that we"
```

```
***** Listing of Program 'LESSON3' *****                    07/10/83 - 01:02:38

2280 PRINT"tested a counter with an IF statement to see if it had reached"
2290 PRINT"a desired level. If it had, then we went to another part of the"
2300 PRINT"program, or STOPped.  There is a set of statements in BASIC"
2310 PRINT"that lets us shorten the statements needed to duplicate this"
2320 PRINT"kind of program.  The set of statements is called FOR - NEXT."
2330 PRINT
2340 PRINT
2350 INPUT"press ENTER";I$
2360 GOSUB 5590
2370 PRINT"                    FOR - NEXT (cont)"
2380 GOSUB 6150
2390 PRINT"The two programs above will print exactly the same output."
2400 PRINT"The COUNTER loop requires more statements and is not as"
2410 PRINT"efficient as the FOR NEXT loop."
2420 PRINT
2430 INPUT"press ENTER";I$
2440 GOSUB 5590
2450 GOSUB 6150
2460 PRINT"Line 10 of the FOR NEXT loop initializes X, the same way that
2470 PRINT"lines 10 and 20 of the COUNTER loop do. The FOR NEXT loop sees"
2480 PRINT"line 10 as 'I'm going to start a loop. X is the counter. I will'"
2490 PRINT"begin with X = 1 and when X is GREATER than 3 I will GOTO the'
2500 PRINT"statement that is after the NEXT X statement'."
2510 PRINT
2520 INPUT"press ENTER";I$
2530 GOSUB 5590
2540 GOSUB 6150
2550 PRINT"Line 30 of the COUNTER loop and line 30 of the FOR NEXT loop'
2560 PRINT"are the same and perform the same function.'
2570 PRINT"Note that the PRINT statement was indented a few spaces in the '
2580 PRINT"FOR NEXT loop.  This lets you see the loop structure better.'
2590 PRINT"You should do that in your own programming, too.'
2600 PRINT
2610 INPUT"press ENTER";I$
2620 GOSUB 5590
2630 GOSUB 6150
2640 PRINT"The IF statement in line 40 of the COUNTER loop determines if X'
2650 PRINT"has reached 3.  This was done automatically by the FOR NEXT"
2660 PRINT"loop because the last value in line 10 was specified as 3.'
2670 PRINT"Line 30 of the FOR NEXT loop is the same as the UNCONDITIONAL"
2680 PRINT"GOTO in line 50 of the COUNTER loop.  The NEXT X statement"
2690 PRINT"INCREMENTS X and sends control to the FOR statement."
2700 INPUT"In this case control goes to line 1.....press ENTER";I$
2710 GOSUB 5590
2720 PRINT"If FOR X = value #1, to value #2"
2730 PRINT"      PRINT X"
```

```
2740 PRINT"30 NEXT X"
2750 PRINT
2760 PRINT"In summar.. the FOR X = (value #1) to (value #2)"
2770 PRINT"causes the variable X to be initially set at value #1."
2780 PRINT"and the program executes the next lines until it reaches the"
2790 PRINT"NEXT X statement.  (X is an example. any VARIABLE works). The"
2800 PRINT"NEXT statement causes an UNCONDITIONAL GOTO to the FOR state-"
2810 PRINT"ment.  The variable is incremented and the FOR word tests to "
2820 PRINT see if it EXCEEDS value #2.  IF it does, then control passes"
2830 PRINT"to the statement that follows the NEXT statement."
2840 PRINT"If there are no statements passed the NEXT word, the program"
2850 PRINT"ENDs."
2860 PRINT
2870 INPUT"press ENTER";T$
2880 GOSUB 5590
2890 PRINT"                    FOR NEXT (cont)"
2900 PRINT
2910 PRINT"This concludes the initial FOR NEXT section.  You should under-"
2920 PRINT"stand what a simple FOR NEXT statement does.  If you do not,"
2930 PRINT"you will be given a chance to review this section before you"
2940 PRINT"go on.  But before that - QUIZ time!"
2950 PRINT
2960 INPUT"press ENTER";T$
2970 GOSUB 5590
2980 FOR X = 1 TO 3
2990    PRINT X*2
3000 NEXT X
3010 PRINT
3020 PRINT"Which of the following programs made the above list?"
3030 PRINT
3040 PRINT"A  10 FOR X=1 TO 3          B  10 FOR I = 1 TO 4"
3050 PRINT"   20    PRINT X*2             20    PRINT I*2"
3060 PRINT"   30 NEXT X                   30 NEXT I"
3070 PRINT
3080 PRINT"C  10 X=X-1                 D  10 Z=Z+1"
3090 PRINT"   20 print X*2                20 print Z*2"
3100 PRINT"   30 IF X=3 THEN GOTO 10      30 IF Z=4 THEN STOP"
3110 PRINT"   40 GOTO 10                  40 GOTO 10"
3120 PRINT
3130 INPUT"press the letter opposite the correct choice and press ENTER";T$
3140 PRINT
3150 IF T$ = "A" THEN GOTO 3170
3160 PRINT"WRONG - the correct answer is A"
3180 GOTO 3190
3170 PRINT"CORRECT"
3190 PRINT
```

```
3190 INPUT"press ENTER";T$
3200 GOSUB 5590
3210 PRINT"The NEXT statement is the same as an UNCONDITIONAL BRANCHING"
3220 PRINT"Statement.  It branches to the next line under the FOR state-"
3230 PRINT"ment."
3240 PRINT
3250 PRINT"Is the paragraph above TRUE or FALSE?"
3260 PRINT
3270 PRINT"A  TRUE"
3280 PRINT"B  FALSE"
3290 PRINT
3300 INPUT"ENTER the letter opposite the correct answer";T$
3310 PRINT
3320 IF T$ = "B" THEN GOTO 3370
3330 PRINT"WRONG - the NEXT statement is UNCONDITIONAL, but it branches"
3340 PRINT"        to the SAME line as the FOR statement.  It also incre-"
3350 PRINT"        ments the variable before the FOR statement tests it."
3360 GOTO 3380
3370 PRINT"CORRECT"
3380 PRINT
3390 INPUT"press ENTER";T$
3400 GOSUB 5590
3410 IF T$ = "B" THEN GOTO 2240
3420 RETURN
3430 GOSUB 5590
3440 PRINT"                    Advanced FOR NEXT"
3450 PRINT
3460 PRINT"10 FOR X = 1 TO 10 STEP 2"
3470 PRINT"20    PRINT X"
3480 PRINT"30 NEXT X"
3490 PRINT
3500 PRINT"Gives us:"
3510 PRINT
3520 FOR X = 1 TO 10 STEP 2
3530     PRINT X
3540 NEXT X
3550 PRINT
3560 INPUT"Interesting, isn't it ..... press ENTER for an explanation";T$
3570 GOSUB 5590
3580 PRINT"                    Advanced FOR NEXT"
3590 PRINT
3600 PRINT"10 FOR X = 1 TO 10 STEP 2"
3610 PRINT"20    PRINT X"
3620 PRINT"30 NEXT X"
3630 PRINT
3640 PRINT"The STEP word in the FOR statement caused the value of X to"
```

```
3650 PRINT"be incremented by 2 instead of 1.  The output of 1,3,5,7,9 was"
3660 PRINT"correct because the NEXT statement is the statement that "
3670 PRINT"actually increments the value, so the first time through,"
3680 PRINT"X was equal to 1, the next time through it was equal to 3, etc."
3690 PRINT
3700 INPUT"press ENTER";T$
3710 GOSUB 5590
3720 PRINT "                    Advanced FOR NEXT"
3730 PRINT
3740 PRINT"10 FOR X = 10 to 1 STEP -2"
3750 PRINT"20   PRINT X"
3760 PRINT"30 NEXT X"
3770 PRINT
3780 PRINT"Gives us:"
3790 PRINT
3800 FOR X = 10 TO 1 STEP -2
3810     PRINT X
3820 NEXT X
3830 PRINT
3840 INPUT"press ENTER for explanation";T$
3850 GOSUB 5590
3860 PRINT"10 FOR X = 10 to 1 STEP -2"
3870 PRINT"20   PRINT X"
3880 PRINT"30 NEXT X"
3890 PRINT
3900 PRINT"In this case, we STEPped DOWN instead of up. Notice that the"
3910 PRINT"first value of the FOR statement is the largest value.  It"
3920 PRINT"would not make sense to specify a negative STEP and give the"
3930 PRINT"range of a positive STEP. For example:"
3940 PRINT
3950 PRINT"10 FOR X = 1 to 10 STEP -1"
3960 PRINT
3970 PRINT"This doesn't make sense, and would not work."
3980 PRINT
3990 INPUT"press ENTER";T$
4000 GOSUB 5590
4010 PRINT"You may include a FOR NEXT statement within another FOR NEXT"
4020 PRINT"statement.  If you do, it is called NESTED looping."
4030 PRINT"For example:"
4040 GOSUB 6290
4050 PRINT"Before we explain the output, do you see how we indented the"
4060 PRINT"statements within the first loop, and then further indented"
4070 PRINT"the statements within the second loop?  It makes them easier"
4080 PRINT"to understand (and explain)."
4090 PRINT
4100 INPUT"press ENTER for output";T$
```

```
4110 GOSUB 5590
4120 PRINT"            Advanced FOR NEXT (NESTED LOOPS)"
4130 PRINT
4140 FOR X = 1 TO 2
4150     FOR Y = 1 TO 2
4160         PRINT X,Y
4170    NEXT Y
4180 NEXT X
4190 PRINT
4200 GOSUB 6290
4210 PRINT"Try to figure how the program produced the columns above it"
4220 INPUT"and then press ENTER";T$
4230 GOSUB 5590
4240 GOSUB 6290
4250 PRINT"On the first pass through the program, the FOR NEXT loop for"
4260 PRINT"Y was set up, and control passed to the lines between FOR X ="
4270 PRINT"1 to 2, and NEXT X.  Those lines happened to be another loop"
4280 PRINT"with the variable Y as the FOR NEXT variable.  When the Y FOR"
4290 PRINT"NEXT executed the first time, X was equal to 1 and Y was"
4300 PRINT"printed twice, once as a 1 and once as a 2. When the Y loop"
4310 PRINT"finished, the NEXT X statement caused control to GOTO line"
4320 PRINT"10 again.  The process was repeated, but X now was equal to 2."
4330 PRINT
4340 INPUT"press ENTER";T$
4350 GOSUB 5590
4360 FOR X = 1 TO 2
4370     FOR Y = 1 TO 2
4380         PRINT X,Y
4390     NEXT Y
4400 NEXT X
4410 PRINT
4420 PRINT"10 FOR X = 1 TO 2"
4430 PRINT"20     FOR Y = 1 TO 2"
4440 PRINT"30         PRINT X,Y"
4450 PRINT"40     NEXT Y"
4460 PRINT"50 NEXT X"
4470 PRINT
4480 INPUT"Can you figure it out now?  Think about it and then press ENTER";T$
4490 GOSUB 5590
4500 GOSUB 6290
4510 PRINT"The most common mistake that programmers make when using nested"
4520 PRINT"loops, is mislabeling the NEXT statements.  Notice that the"
4530 PRINT"NEXT statement for the Y variable is placed before the NEXT X"
4540 PRINT"statement."
4550 PRINT
4560 INPUT"press ENTER";T$
```

```
4570 GOSUB 5590
4580 PRINT"The format for nesting loops is:"
4590 PRINT
4600 PRINT"LOOP 1 (for)"
4610 PRINT"       LOOP 2 (for)"
4620 PRINT"             LOOP 3 (for)"
4630 PRINT"             LOOP 3 (next)"
4640 PRINT"       LOOP 2 (next)"
4650 PRINT"LOOP 1 (next)"
4660 PRINT
4670 PRINT"Notice that you must back out of a nest in reverse order."
4680 PRINT
4690 INPUT"press ENTER";T$
4700 GOSUB 5590
4710 PRINT"Is the following program valid?"
4720 PRINT
4730 PRINT"10 FOR X = 1 to 200"
4740 PRINT"20     FOR I = 1 to 2"
4750 PRINT"30         PRINT X+Z"
4760 PRINT"40     NEXT Z"
4770 PRINT"50 NEXT X"
4780 PRINT
4790 PRINT"A  YES it is a good program"
4800 PRINT"B  NO it is not a valid program"
4810 PRINT
4820 INPUT"ENTER the letter opposite the correct answer";T$
4830 PRINT
4840 IF T$ = "A" GOTO 4880
4850 PRINT"WRONG - the correct answer is A.  There is nothing wrong"
4860 PRINT"        with the program."
4870 GOTO 4900
4880 PRINT"CORRECT"
4890 PRINT
4900 INPUT"press ENTER";T$
4910 GOSUB 5590
4920 PRINT"Give the first statement of a FOR NEXT loop, that will"
4930 PRINT"cause the loop to increment the variable I from one to 20 in"
4940 PRINT"steps of 2.  Use 10 as your line number and leave one space"
4950 PRINT"between all words, numbers, and variables."
4960 PRINT
4970 INPUT"ENTER your answer";T$
4980 PRINT
4990 IF T$ = "10 FOR I = 1 TO 20 STEP 2" GOTO 5030
5000 PRINT"WRONG - the answer is:"
5010 PRINT"        10 FOR I = 1 TO 20 STEP 2"
5020 GOTO 5040
```

```
5030 PRINT"CORRECT! - You are a winner!"
5040 PRINT
5050 INPUT"press ENTER";T$
5060 GOSUB 5590
5070 PRINT"Is the following program valid?"
5080 PRINT
5090 PRINT"10 FOR X = 1 TO 2"
5100 PRINT"20     FOR I = 1 TO 5"
5110 PRINT"30         PRINT "CHR$(34)"This is the last question"CHR$(34)
5120 PRINT"40     NEXT I"
5130 PRINT"50 NEXT X"
5140 PRINT
5150 PRINT"A   Yes, it is valid"
5160 PRINT"B   No, it is not valid"
5170 PRINT
5180 INPUT"ENTER the letter opposite the correct answer";T$
5190 PRINT
5200 IF T$ = "A" GOTO 5230
5210 PRINT"WRONG - the program is valid"
5220 GOTO 5310
5230 GOSUB 5590
5240 FOR X = 1 TO 2
5250     FOR Y = 1 TO 5
5260         PRINT"This is the last question"
5270     NEXT Y
5280 NEXT X
5290 PRINT
5300 PRINT"CORRECT!  -  and the output is shown above"
5310 PRINT
5320 INPUT"press ENTER";T$
5330 GOSUB 5590
5340 PRINT"Remember in the beginning of this section when we mentioned"
5350 PRINT"the program that produced three columns of figures, the first"
5360 PRINT"column was integer 1 through 8, the second column was the "
5370 PRINT"square of the first, and the third the square of the second?"
5380 PRINT"Think you can figure out how we did it?"
5390 PRINT
5400 INPUT"press ENTER for the answer";T$
5410 GOSUB 5590
5420 FOR X = 1 TO 8
5430     PRINT X, X*X, (X*X)+(X*X)
5440 NEXT X
5450 PRINT
5460 PRINT"10 FOR X = 1 TO 8"
5470 PRINT"20     PRINT X,X*X,(X*X)+(X*X)"
5480 PRINT"30 NEXT X
```

```
5490 PRINT
5500 PRINT"It looks like child's play now, doesn't it?"
5510 PRINT
5520 INPUT"press ENTER";T$
5530 GOSUB 5800
5540 IF T$ = "B" THEN GOTO 3430
5550 RETURN
5560 REM **
5570 REM ** This subroutine clears the screen on any terminal
5580 REM **
5590 FOR X = 1 TO 24
5600    PRINT
5610 NEXT X
5620 RETURN
5630 REM **
5640 REM ** THIS ROUTINE IS THE MENU
5650 REM **
5660 PRINT"                    LESSON 3"
5670 PRINT
5680 PRINT"This is the first part of a two part lesson"
5690 PRINT"It is divided into the following sections."
5700 PRINT
5710 PRINT"1)  Introduction & KILL     4)  FOR NEXT Statements"
5720 PRINT"2)  LOOPs (Intro)           5)  Advanced FOR NEXT"
5730 PRINT"3)  COUNTER variables"
5740 PRINT
5750 PRINT
5760 RETURN
5770 REM **
5780 REM ** THIS LETS STUDENT REVIEW LESSONS IF HE WISHES
5790 REM **
5800 GOSUB 5590
5810 PRINT"Which do you wish to do?"
5820 PRINT
5830 PRINT"A  Continue on"
5840 PRINT"B  Review this section again"
5850 PRINT
5860 INPUT"press the letter opposite the correct answer and press ENTER";T$
5870 IF T$ = "A" OR T$ = "B" THEN RETURN
5880 GOTO 5800
5890 REM **
5900 REM ** THIS IS EXAMPLE PROGRAM TO SHOW HOW A LOOP WORKS
5910 REM **
5920 PRINT
5930 PRINT"10 REM  This program calculates a table of squares of numbers"
5940 PRINT"20 X = 0"
```

```
5950 PRINT"30 X = X + 1"
5960 PRINT"40 PRINT X,X*X,(X*X)*(X*X)"
5970 PRINT"50 IF X = 8 THEN GOTO 70"
5980 PRINT"60 GOTO 30"
5990 PRINT"70 END"
6000 PRINT
6010 RETURN
6020 REM
6030 REM This is example program for COUNTER section
6040 REM
6050 PRINT
6060 PRINT"10 Z=0"
6070 PRINT"20 Z=Z+1"
6080 PRINT"30 IF Z = 10 THEN STOP"
6090 PRINT"40 GOTO 20"
6100 PRINT
6110 RETURN
6120 REM
6130 REM This example is for the FOR NEXT section
6140 REM
6150 PRINT
6160 PRINT"COUNTER loop               FOR NEXT loop"
6170 PRINT
6180 PRINT"10 X=0                       10 FOR X = 1 to 8"
6190 PRINT"20 X=X+1                     20     PRINT X"
6200 PRINT"30 PRINT X                   30 NEXT X"
6210 PRINT"40 IF X = 8 GOTO 60          40 END"
6220 PRINT"50 GOTO 20"
6230 PRINT"60 END"
6240 PRINT
6250 RETURN
6260 REM
6270 REM This example is for the Advanced FOR NEXT section
6280 REM
6290 PRINT
6300 PRINT"10 FOR X = 1 TO 2"
6310 PRINT"20     FOR Y = 1 TO 2"
6320 PRINT"30         PRINT X,Y"
6330 PRINT"40     NEXT Y
6340 PRINT"50 NEXT X"
6350 PRINT
6360 RETURN
6370 RUN "MENU"
6380 RUN"LESSON3A"
6390 END
```

```
250 GOSUB 4080
260 GOSUB 4150
270 PRINT"A  I'm taking this part in its entirety."
280 PRINT"B  I wish to review selected areas. (or take the test)"
290 PRINT"C  I want to go to the first part."
300 PRINT"D  I want to return to the Menu."
310 PRINT
320 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
330 IF T$ = "D" GOTO 4750
340 IF T$ = "C" GOTO 4760
350 IF T$ = "B" GOTO 430
360 IF T$ <>"A" GOTO 270
370 GOSUB 540
380 GOSUB 1360
390 GOSUB 2430
400 GOSUB 3960
410 GOSUB 4080
420 GOTO 4760
430 GOSUB 4080
440 GOSUB 4150
450 PRINT
460 PRINT"Please type in the number beside the area you wish"
470 PRINT"to review (1 through 4) and then press ENTER - press 0 and"
480 PRINT"press ENTER to return to the Menu."
490 PRINT
500 INPUT"What is your choice";N
510 IF N = 0 GOTO 4750
520 ON N GOSUB 540  ,1360 ,2430 ,3960
530 GOTO 430
540 GOSUB 4080
550 PRINT"                          Arrays  (Intro)"
560 PRINT
570 PRINT"We have seen that variables are storage places for data.  In "
580 PRINT"large programs, it is difficult to manipulate large data bases"
590 PRINT"without having numerous variables to assign the bits of data"
600 PRINT"to.  However, there is a way to group our variables into"
610 PRINT"similar bunches that makes it easier for us to tell what part"
620 PRINT"of the data base our variable belongs to.  We can use"
630 PRINT"ARRAYs."
632 PRINT
634 INPUT"press ENTER";T$
636 GOSUB 4080
637 PRINT"                          Arrays (Intro)"
638 PRINT
639 PRINT"One use for ARRAYs would be to make a training program that"
640 PRINT"listed the people on training, their time in the organization."
```

```
650 PRINT"or rank, and their training status.  We could group the major"
660 PRINT"catagories (name, rank, training status) into three variables"
670 PRINT"and use subscripts to provide a place for each entry in our"
680 PRINT"data base."
690 PRINT
700 INPUT"press ENTER";T$
710 GOSUB 4080
720 GOSUB 4410
730 PRINT"We could assign subscripted ARRAYs  to the three main cata-"
740 PRINT"gories.  An ARRAY has the following format:"
750 PRINT
760 PRINT"     Variable (subscript)"
770 PRINT
780 PRINT"The subscript is enclosed in parenthesis.  Examples of valid"
790 PRINT"ARRAY variables are: N$(1), R$(2), T(9)"
800 PRINT
810 INPUT"press ENTER";T$
820 GOSUB 4080
830 GOSUB 4410
840 PRINT"We could use our ARRAYs to hold the above data."
850 PRINT"We could use N$(0), N$(1), and N$(2) to indicate the three"
860 PRINT"names, R$(0), R$(1), and R$(2) to indicate the three ranks,"
870 PRINT"and T(0), T(1), and T(2) to represent the three training"
880 PRINT"levels.  Note that 0 is a valid subscript."
890 PRINT
900 PRINT
910 INPUT"press ENTER";T$
920 GOSUB 4080
930 PRINT"10 N$(0) = "CHR$(34)"John Doe"CHR$(34)
940 PRINT"20 N$(1) = "CHR$(34)"Jake Robinski"CHR$(34)
950 PRINT"30 N$(2) = "CHR$(34)"Mark Muffin"CHR$(34)
960 PRINT"40 R$(0) = "CHR$(34)"Foreman"CHR$(34)
970 PRINT"50 R$(1) = "CHR$(34)"Peon"CHR$(34)
980 PRINT"60 R$(2) = "CHR$(34)"Specialist"CHR$(34)
990 PRINT"70 T(0)  = 9"
1000 PRINT"80 T(1)  = 3"
1010 PRINT"90 T(2)  = 5"
1020 PRINT"100 PRINT"CHR$(34)"Name              Rank              Training Status"CHR$(34)
1030 PRINT"110 PRINT"
1040 PRINT"120 PRINT N$(0),R$(0),T(0)"
1050 PRINT"130 PRINT N$(1),R$(1),T(1)"
1060 PRINT"140 PRINT N$(2),R$(2),T(2)"
1065 PRINT
1070 INPUT"This program, when RUN, would print our data.....press ENTER";T$
1080 GOSUB 4080
1090 GOSUB 4410
```

```
1100 PRINT"This is what that program would output.  Notice that we have"
1110 PRINT"used only three variables, but we made them ARRAYs so that we"
1120 PRINT"could hold nine bits of data."
1130 PRINT"As we continue through our lessons, we will discover some very"
1140 PRINT"powerful uses for ARRAYs."
1150 PRINT
1160 INPUT"press ENTER";T$
1170 GOSUB 4080
1180 PRINT"Is the following ARRAY and its subscript valid?"
1190 PRINT
1200 PRINT"A(0)"
1210 PRINT
1220 PRINT"A  TRUE"
1230 PRINT"B  FALSE"
1240 PRINT
1250 INPUT"ENTER the letter opposite the correct answer";T$
1260 PRINT
1270 IF T$ = "A" GOTO 1300
1280 PRINT"WRONG - the correct answer is A"
1290 GOTO 1310
1300 PRINT"CORRECT"
1310 PRINT
1320 INPUT"press ENTER";T$
1330 GOSUB 4290
1340 IF T$ = "B" GOTO 540
1350 RETURN
1360 GOSUB 4080
1370 PRINT"                              ARRAYs"
1375 PRINT
1380 PRINT"If you have a little mathematics in your background, you will"
1390 PRINT"have noticed that ARRAYs  are almost the same as their math"
1400 PRINT"equivalent, except that the subscripts  are in parenthesis"
1410 PRINT"instead of slightly lower and to the right of the variable."
1420 PRINT
1430 PRINT"The previous examples all dealt with a ONE-DIMENSIONAL "
1440 PRINT"ARRAY.  That is, there was only one number in parenthesis"
1450 PRINT"that was significant.  ARRAYs with TWO, THREE, FOUR, or more"
1460 PRINT"dimensions are possible.  Most dialects of BASIC, including"
1470 PRINT"Microsoft, will handle at least 8 dimensions. An example of a"
1480 PRINT"TWO dimension ARRAY would be N(2,2).  Notice that the extra"
1490 PRINT"dimension was designated by just adding another subscript in-"
1500 PRINT"side the parenthesis.  A THREE DIMENSIONED ARRAY looks like"
1510 PRINT"this: R(2,1,9) or T$(5,44,3)  (or any combination of numbers)"
1520 INPUT"press ENTER";T$
1530 GOSUB 4030
1540 GOSUB 4520
```

```
1550 PRINT"The above program combines several of the techniques that we"
1560 PRINT"have been learning.  Before we tell you the answer, try to "
1570 PRINT"figure out what the output of the program would be.  We warn"
1580 PRINT"you, it is a little tricky, but see if you can figure it out."
1590 PRINT
1600 INPUT"press ENTER";T$
1610 GOSUB 4080
1620 GOSUB 4520
1630 PRINT"5 10 15 20"
1640 PRINT
1650 PRINT"This is the output.  Lines 10 and 20 are nested FOR NEXT loops."
1660 PRINT"They set up the READ statement in line 30 so that it will READ"
1670 PRINT"in the values that are in the DATA statement and assign the"
1680 PRINT"current X,Y subscript to it. Values are read in one at a time."
1690 INPUT"press ENTER";T$
1700 GOSUB 4080
1710 GOSUB 4520
1720 PRINT"On the first pass, X = 1 and Y = 1, A(1,1) therefore, equals 5"
1730 PRINT"The semi-colon on the end of the print statement causes the "
1740 PRINT"numbers to be printed side by side instead of on separate lines"
1750 PRINT"and they all have one space between them. (caused by the ';')"
1760 PRINT"On the second iteration of Y, Y will equal 2 and X = 1, A(1,2)"
1770 PRINT"will equal 10."
1780 PRINT
1790 INPUT"press ENTER";T$
1900 GOSUB 4080
1810 GOSUB 4520
1820 PRINT"When the second iteration of Y is done, control will pass to "
1830 PRINT"line 10 and X will begin ITs second iteration.  The Y loop"
1840 PRINT"will start all over again and when line 30 is executed (3rd"
1850 PRINT"time), A(2,1) will equal 15. Finally, Y will execute for the"
1860 PRINT"fourth time (second time while X = 2), and A(X,Y) will have all"
1870 PRINT"numbers in the DATA statement, and the numbers will have "
1880 PRINT"printed out."
1885 PRINT
1890 INPUT"press ENTER";T$
1900 GOSUB 4080
1910 GOSUB 4650
1920 PRINT"Another way to look at the ARRAY is to visualize it as above."
1930 PRINT"The ARRAY has four of its pockets loaded (we ignored the 0"
1940 PRINT"pockets so the explanation would be simpler. They are still"
1950 PRINT"there, they are just not used) When A(X,Y) = 10, X must equal 1"
1960 PRINT"and Y must equal 2.  Do you see?"
1965 PRINT
1970 INPUT"press ENTER";T$
1980 GOSUB 4080
```

```
1990 GOSUB 4650
2000 PRINT" If a value of one of the pockets of array A(X,Y) equals 15,"
2010 PRINT"what are the values of X & Y that would reference that pocket?"
2020 PRINT
2030 PRINT"A   X=1, Y=1"
2040 PRINT"B   X=1, Y=2"
2050 PRINT"C   X=2, Y=1"
2060 PRINT"D   X=2, Y=2"
2065 PRINT
2070 INPUT"ENTER the letter opposite the correct answer";T$
2080 PRINT
2090 IF T$ = "C" GOTO 2120
2100 PRINT"WRONG - the correct answer is C"
2110 GOTO 2130
2120 PRINT"CORRECT !   *** OUTSTANDING ***"
2130 PRINT
2140 INPUT"press ENTER";T$
2150 GOSUB 4080
2160 GOSUB 4650
2170 PRINT"If X = 1 and Y = 1, what value is in the pocket pointed to by"
2180 PRINT"ARRAY A(X,Y)?"
2190 PRINT
2200 PRINT"A  10          B  15"
2210 PRINT"C  20          D   5"
2220 PRINT
2230 INPUT"ENTER the letter opposite the correct answer";T$
2240 PRINT
2250 IF T$ = "D" GOTO 2280
2260 PRINT"WRONG - the correct answer is D"
2270 GOTO 2290
2280 PRINT"CORRECT !"
2290 PRINT
2300 INPUT"press ENTER";T$
2310 GOSUB 4080
2320 GOSUB 4650
2330 PRINT"The FIRST subscript in an ARRAY ALWAYS indicates the ROWs of"
2340 PRINT"DATA, and the SECOND subscript indicates the COLUMNs of data."
2350 PRINT"Therefore, ARRAY A(X,Y) has X rows and Y columns.  You will"
2360 PRINT"find that once you visualize a two dimensional ARRAY, the "
2370 PRINT"others will come quite easily."
2380 PRINT
2390 INPUT"press ENTER";T$
2400 GOSUB 4290
2410 IF T$ = "B" GOTO 1360
2420 RETURN
2430 GOSUB 4080
```

```
2440 PRINT"                    Dimension & Clear"
2450 PRINT
2460 PRINT"Whenever you use an ARRAY in BASIC, the processor has to make"
2470 PRINT"room for all the extra memory pockets that you will use."
2480 PRINT"Up to a limit, the machine can handle unexpected ARRAYs, but"
2490 PRINT"after you designate more than 10 pockets or 3 dimensions, the"
2500 PRINT"computer must know ahead of time so it can reserve enough"
2510 PRINT"space.  The way you tell the computer to use an ARRAY"
2520 PRINT"with at least one subscript bigger than 10 or with more than 3"
2530 PRINT"dimensions to it (ie N(3,3,3,3)), you use the"
2540 PRINT"DIM statement.  DIM stands for DIMENSION, and it must be "
2550 PRINT"used before the ARRAY is used, and it cannot be changed once"
2560 PRINT"the program is RUNning."
2570 PRINT
2580 INPUT"press ENTER";T$
2590 GOSUB 4080
2600 PRINT"10 DIM B(12)"
2610 PRINT"20 FOR X = 1 TO 12"
2620 PRINT"30     READ B(X)"
2630 PRINT"40 NEXT X"
2640 PRINT"50 DATA 5,10,15,20,25,30,35,40,45,50,55,60"
2650 PRINT"60 FOR X = 1 TO 12"
2660 PRINT"70     PRINT B(X);"
2670 PRINT"80 NEXT X"
2680 PRINT"RUN"
2690 PRINT
2700 PRINT"5 10 15 20 25 30 35 40 45 50 55 60"
2710 PRINT
2720 PRINT"The DIM statement told the computer that 12 pockets were needed"
2730 PRINT"and the READ, DATA, and PRINT statements filled the ARRAY and"
2740 PRINT"printed it out."
2745 PRINT
2750 INPUT"press ENTER";T$
2760 GOSUB 4080
2770 PRINT"                    Dimension and Clear"
2775 PRINT
2780 PRINT"Another problem the computer has with memory allocation, is"
2790 PRINT"reserving enough room for STRING space.  For every letter in"
2800 PRINT"a STRING, the BASIC processor must use a little over one memory"
2810 PRINT"location. (it must be able to find the letter once it stores it"
2820 PRINT"so it uses one location to store the letter, and another to"
2830 PRINT"remind it where it put the STRING in the first place!)"
2840 PRINT"Unlike number variables, STRINGs can use up to 255 characters"
2850 PRINT"per line (numbers seldom use more than 4 - the reason is rather"
2860 PRINT"technical, you may wish to look up how data is handled intern-"
2870 PRINT"ally by the computer in a reference book).  If you are going"
```

```
2880 PRINT"to use more than 50 characters worth of STRING space, you must"
2890 PRINT"CLEAR more room for it.  The clear statement actually WIPES OUT"
2900 PRINT"data space and reserves memory locations, so it must ALWAYS be"
2910 PRINT"the first statement if you are going to need it."
2915 PRINT
2920 INPUT"press ENTER";T$
2930 GOSUB 4080
2940 PRINT"                    Dimension and Clear"
2950 PRINT
2960 PRINT"If you use the CLEAR statement in the middle of a program,"
2970 PRINT"the accumulation of data that you have stored in variables to"
2980 PRINT"the point that the CLEAR word was used, will be zeroed out."
2990 PRINT
3000 PRINT"10 X = 150"
3010 PRINT"20 CLEAR"
3020 PRINT"30 PRINT X"
3030 PRINT"RUN"
3040 PRINT
3050 PRINT"0"
3060 PRINT
3070 INPUT"Do you see? The CLEAR word zeroed out X'.....press ENTER";T$
3080 GOSUB 4080
3090 PRINT"10 CLEAR 1000"
3100 PRINT"20 DIM G(50), A$(100)"
3110 PRINT"30 FOR X = 1 TO 100"
3120 PRINT"40     READ A$(X)"
3130 PRINT".......etc."
3140 PRINT
3150 PRINT"The above shows the first 4 lines of a program that is going"
3160 PRINT"to use more than 1000 spaces of string space (that will give"
3170 PRINT"an average of 10 letters per pocket of A$(X)) and is going to"
3180 PRINT"use 50 pockets of the numeric array G. (note that you always"
3190 PRINT"get pocket 0 for free, it is normally not used)"
3200 PRINT
3210 PRINT"Notice the position of the CLEAR and DIM statements."
3220 INPUT"press ENTER";T$
3230 GOSUB 4080
3240 PRINT"What will be the output of the  following program?"
3250 PRINT
3260 PRINT"10 A(13) = 5"
3270 PRINT"20 CLEAR"
3280 PRINT"30 PRINT A(13)"
3290 PRINT
3300 PRINT"A  0 because the clear statement is in the wrong place."
3310 PRINT
3320 PRINT"B  5  - the clear statement only affects STRINGS."
```

```
3330 PRINT
3340 PRINT"C  Nothing because A(13) will cause an ERROR"
3350 PRINT
3360 INPUT"ENTER the letter opposite the correct answer";T$
3370 PRINT
3380 IF T$ = "C" THEN GOTO 3410
3390 PRINT"WRONG - C is the right answer"
3400 GOTO 3420
3410 PRINT"CORRECT - GoooooOOOOOOOD JOB!"
3420 PRINT
3430 INPUT"press ENTER";T$
3440 GOSUB 4080
3450 PRINT"The CLEAR word is used to clear storage space for strings, but"
3460 PRINT"it also wipes out other data."
3470 PRINT
3480 PRINT"A   TRUE"
3490 PRINT"B   FALSE"
3500 PRINT
3510 INPUT"ENTER the letter opposite the correct answer";T$
3520 PRINT
3530 IF T$ = "A" GOTO 3560
3540 PRINT"WRONG - the correct answer is A"
3550 GOTO 3570
3560 PRINT"CORRECT - GoooooooooooOOOOOOOOOOOOOOOOOOD JOB!"
3570 PRINT
3580 INPUT"press ENTER";T$
3590 GOSUB 4080
3600 PRINT"As a reminder, the DIM statement does not create the ARRAY, it "
3610 PRINT"only defines its size.  It may be used to MINIMIZE storage"
3620 PRINT"space by designating ARRAYs that are LESS than 10.  That way"
3630 PRINT"the computer will not automatically reserve more space than"
3640 PRINT"needed.  For example:"
3650 PRINT
3660 PRINT"10 DIM A(2,2)"
3670 PRINT
3680 PRINT"This would be a valid, and memory conserving statement. The "
3690 PRINT"processor wouldn't reserve a 10 X 10 pocket ARRAY for A(X,X), it"
3700 PRINT"would only reserve a 2 X 2."
3710 PRINT
3720 INPUT"press ENTER";T$
3730 GOSUB 4080
3740 PRINT"What is wrong with this program?"
3750 PRINT
3760 PRINT"10 CLEAR 1000"
3770 PRINT"20 DIM A(2)"
3780 PRINT"30 A(2) = 5"
```

```
3790 PRINT"40 PRINT A(2)"
3800 PRINT
3810 PRINT"A   The DIM statement is in the wrong place."
3820 PRINT"B   The CLEAR statement is invalid."
3830 PRINT"C   Nothing."
3840 PRINT
3850 INPUT"ENTER the letter opposite the correct answer";T$
3860 PRINT
3870 IF T$ = "C" GOTO 3900
3880 PRINT"WRONG - the correct answer is C"
3890 GOTO 3910
3900 PRINT"CORRECT"
3910 PRINT
3920 INPUT"press ENTER";T$
3930 GOSUB 4290
3940 IF T$ = "B" GOTO 2430
3950 RETURN
3960 GOSUB 4080
3970 PRINT"                              TEST"
3980 PRINT
3990 PRINT"You have completed this lesson.  ENTER a 'C' to Continue"
4000 PRINT"to the TEST or ENTER an 'R' to start over."
4010 PRINT
4020 INPUT"ENTER your choice";T$
4030 IF T$ = "C" THEN GOTO 4740
4040 RUN
4050 REM **
4060 REM ** This subroutine clears the screen on any terminal
4070 REM **
4080 FOR X = 1 TO 24
4090    PRINT
4100 NEXT X
4110 RETURN
4120 REM **
4130 REM ** THIS IS THE MENU SUBROUTINE
4140 REM **
4150 PRINT"                    LESSON 3B"
4160 PRINT
4170 PRINT"This is the second part of a two part lesson"
4180 PRINT"It is divided into the following sections."
4190 PRINT
4200 PRINT"1)  Arrays (Introduction)  3)  Dimension & Clear"
4210 PRINT"2)  Arrays                 4)  Test"
4220 PRINT
4230 PRINT
4240 PRINT
```

```
4250 RETURN
4260 REM **
4270 REM ** THIS LETS STUDENT REVIEW LESSONS AGAIN
4280 REM **
4290 GOSUB 4080
4300 PRINT"Which do you wish to do?"
4310 PRINT
4320 PRINT"A  Continue on"
4330 PRINT"B  Review this section again"
4340 PRINT
4350 INPUT"press the letter opposite the correct answer and press ENTER";T$
4360 IF T$ = "A" OR T$ = "B" THEN RETURN
4370 GOTO 4290
4380 REM
4390 REM This subroutine is for the first Subscripts example
4400 REM
4410 PRINT
4420 PRINT"Name                    Rank          Training Status"
4430 PRINT
4440 PRINT"John Doe                Foreman             9"
4450 PRINT"Jake Robinski           Peon                3"
4460 PRINT"Mark Muffin             Specialist          5"
4470 PRINT
4480 RETURN
4490 REM **
4500 REM ** THIS ROUTINE IS FOR ARRAYS EXAMPLE
4510 REM **
4520 PRINT
4530 PRINT"10 FOR X = 1 TO 2"
4540 PRINT"20     FOR Y = 1 TO 2"
4550 PRINT"30         READ A(X,Y)"
4560 PRINT"40         PRINT A(X,Y);"
4570 PRINT"50     NEXT Y"
4580 PRINT"60 NEXT X"
4590 PRINT"70 DATA 5,10,15,20"
4600 PRINT
4610 RETURN
4620 REM **
4630 REM ** THIS ROUTINE IS 2ND ARRAY EXAMPLE
4640 REM **
4650 PRINT"                    Y"
4660 PRINT"                 1.     2."
4670 PRINT"              I-----I-----I"
4680 PRINT"           1. I  5  I 10  I"
4690 PRINT"        X     I-----I-----I       ARRAY A(X,Y)"
4700 PRINT"           2. I  15 I  20 I"
```

```
4710 PRINT"                I-----I-----I"
4720 PRINT
4730 RETURN
4740 RUN "TEST3"
4750 RUN "MENU"
4760 RUN"LESSON3"
4770 END
```

```
1000 REM **
1010 REM ** LESSON:  TEST3                VERSION: 1 AUG 83
1020 REM ** AUTHOR:  CAPT DAN CREAGAN
1030 REM **          AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM **            N$(X) = NAMES ARRAY, USED TO READ IN SEQ-
1070 REM **                    UENTIAL NAMES, AND TO WRITE OUT
1080 REM **                    UPDATE NAMES.
1090 REM **            S(X)  = SCORES ARRAY - USED TO READ AND
1100 REM **                    WRITE SCORES
1110 REM **            Q(X)  = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM **                    CORRECT ANSWERS.  IF AN ARRAY
1130 REM **                    ELEMENT EQUALS 1, THE ANSWER WAS
1140 REM **                    CORRECT
1150 REM **
1160 CLEAR 3000
1170 GOSUB 4080
1180 DIM N$(1000)
1190 DIM Q(10)
1200 DIM S(1000)
1210 PRINT"                    FINAL TEST (lesson 3)"
1220 PRINT
1230 PRINT"This test consists of 10 questions. You must get 70 percent"
1240 PRINT"of them correct to pass.  (that's 7 right out of the 10 ques-"
1250 PRINT"tions).  Use only capital letters in your answers, don't"
1260 PRINT"include extra spaces or letters.  GOOD LUCK"
1270 PRINT
1280 INPUT"press ENTER to continue";I$
1290 GOSUB 408.
1300 PRINT"which of the following is valid:"
1310 PRINT
1320 PRINT"A  KILL "CHR$(34)"SLEAZY"CHR$(34)
1330 PRINT"B  KILL SLEAZY"
1340 PRINT"C  UNSAVE "CHR$(34)"SLEAZ:"CHR$(34)
1350 PRINT"D  UNSAVE SLEAZY"
1360 PRINT
1370 INPUT"ENTER the letter opposite the correct answer";I$
1380 PRIN.
1390 IF I$ = "A" GOTO 1450
1400 PRINT"WRONG - the correct answer is A"
1410 PRINT"        Answer B needs SLEAZY in quotes to be right."
1420 PRINT"        If answers C & D UNSAVE is not a BASIC word."
1430 PRINT"        See part 1, 'KILL'"
1440 GOTO 1470
1450 PRINT"CORRECT
```

```
1460 G(1) = 1
1470 PRINT
1480 INPUT"press ENTER";T$
1490 GOSUB 4030
1500 PRINT"Does the FOR NEXT combination have to be used to set up a loop"
1510 PRINT
1520 PRINT"A  YES"
1530 PRINT"B  NO"
1540 PRINT
1550 INPUT"ENTER the letter opposite the correct answer";T$
1560 PRINT
1570 IF T$ = "B" THEN GOTO 1630
1580 PRINT"WRONG - the correct answer is B"
1590 PRINT"        A loop can be made with a variety of techniques,"
1600 PRINT"        including counters and IF statements, FOR - NEXT, etc."
1610 PRINT"        See part 1, LOOPs, FOR NEXT."
1620 GOTO 1650
1630 PRINT"CORRECT"
1640 G(2) = 1
1650 PRINT
1660 INPUT"press ENTER";T$
1670 GOSUB 4030
1680 PRINT"Write out the first line of a FOR NEXT loop using I as the"
1690 PRINT"variable, start the loop at 1 and end it at 5, use a STEP of 2."
1700 PRINT"Use line number 50.  Use all caps, leave one space between all"
1710 PRINT"terms."
1720 PRINT
1730 INPUT"ENTER your answer";T$
1740 PRINT
1750 IF T$ ="50 FOR I = 1 TO 5 STEP 2" THEN GOTO 1790
1760 PRINT "WRONG - the correct answer is 50 FOR I = 1 TO 5 STEP 2"
1770 PRINT"        See part 1, FOR - NEXT."
1780 GOTO 1810
1790 PRINT"CORRECT"
1800 G(3) = 1
1810 PRINT
1820 INPUT"press ENTER";T$
1830 GOSUB 4030
1840 PRINT"10 FOR X = 1 TO 5 STEP 2"
1850 PRINT"20    PRINT X;"
1860 PRINT"30 NEXT X"
1870 PRINT"RUN"
1880 PRINT
1890 PRINT"ENTER the output from this program, leave one space between"
1900 PRINT"terms.   (hint: be sure to consider the semi-colon in line 20)"
1910 PRINT
```

```
1920 INPUT"ENTER your answer";T$
1930 PRINT
1940 IF T$ = "1 3 5" THEN GOTO 2030
1950 PRINT"WRONG - the correct answer is 1 3 5"
1960 PRINT"        The first time through, a '1' is printed, then the"
1970 PRINT"        NEXT X statement increments X by 2. The second time"
1980 PRINT"        through a '3' is printed, BESIDE the 1 (because of the ';')"
1990 PRINT"        then the NEXT X statement again"
2000 PRINT"        increments X by 2. The last time through, a 5 is printed."
2010 PRINT"        See part 1, Advanced FOR - NEXT"
2020 GOTO 2050
2030 PRINT"CORRECT"
2040 B(4) = 1
2050 PRINT
2060 INPUT"press ENTER";T$
2070 GOSUB 4080
2080 PRINT"How many pockets (or elements) can an array have without a"
2090 PRINT"DIM statement?"
2100 PRINT
2110 PRINT"A  10  not counting the 0 element"
2120 PRINT"B  11  not counting the 0 element"
2130 PRINT"C   3  not counting the 0 element"
2140 PRINT"D   4  not counting the 0 element"
2150 PRINT
2160 INPUT"ENTER the letter opposite the correct answer";T$
2170 PRINT
2180 IF T$ = "A" GOTO 2220
2190 PRINT"WRONG - the correct answer is A"
2200 PRINT"        See part 2, ARRAYS"
2210 GOTO 2240
2220 PRINT"CORRECT"
2230 B(5) = 1
2240 PRINT
2250 INPUT"press ENTER";T$
2260 GOSUB 4080
2270 PRINT"How many pockets are there in a 2 X 3 array?"
2280 PRINT
2290 INPUT"ENTER your answer";T$
2300 PRINT
2310 IF T$ = "6" THEN GOTO 2360
2320 PRINT"WRONG - the correct answer is 6"
2330 PRINT         The easy way to count them is to multiply them."
2340 PRINT"        See part 2, ARRAYS."
2350 GOTO 2380
2360 PRINT"CORRECT"
2370 B(6) = 1
```

```
230 PRINT
240 INPUT"press ENTER";T$
250 GOSUB 4080
260 PRINT "How many ROWS does the following array have?"
270 PRINT
280 PRINT"              A(10,5)"
290 PRINT
300 INPUT"ENTER your answer";T$
310 PRINT
320 IF T$ = "1" THEN GOTO 2520
330 PRINT"WRONG - the correct answer is 10"
340 PRINT"       The ROWs are the first subscript of the array."
350 PRINT"       See part 2. ARRAYs."
360 GOTO 2540
370 PRINT"CORRECT
380 Q(7) = 1
390 PRINT
400 INPUT"press ENTER";T$
410 GOSUB 4080
420 PRINT Does the following array need to be DIMensioned?"
430 PRINT
440 PRINT"A(2,3,2,2)"
450 PRINT
460 INPUT"ENTER YES or NO";T$
470 PRINT
480 IF T$ = "YES" THEN GOTO 2680
490 PRINT"WRONG the correct answer is YES"
500 PRINT"     Any array with more than 3 subscripts must be DIMed."
510 PRINT"     See part 2. ARRAYs."
520 GOTO 2700
530 PRINT"CORRECT"
540 Q(8) = 1
550 PRINT
560 INPUT"press ENTER";T$
570 GOSUB 4080
580 PRINT"Which of the following is valid'
590 PRINT
600 PRINT"A  A$(1)"
610 PRINT"B   A$1)"
620 PRINT"C  A$ A$,2)"
630 PRINT"D   A( 1,1$)"
640 PRINT
650 INPUT"ENTER the letter opposite the correct answer";T$
660 PRINT
670 IF T$ =  A" THEN GOTO 1890
680 PRINT"WRONG - the correct answer is A"
```

```
2840 PRINT"       Answer B has parens in the wrong place, C & D tr
2850 PRINT"       to use a string instead of a number for a pocket"
2860 PRINT"       designator."
2870 PRINT"       See part 2, ARRAYs."
2880 GOTO 2910
2890 PRINT"CORRECT"
2900 Q(9) = 1
2910 PRINT
2920 INPUT"press ENTER";T$
2930 GOSUB 4080
2940 PRINT"The CLEAR statement is used for clearing STRING space and "
2950 PRINT"ALWAYs must be used if your STRING use is greater than 100"
2960 PRINT"but doesn't have to be used if your STRING will be LESS than"
2970 PRINT"100 characters."
2980 PRINT
2990 PRINT"Is the above question TRUE or FALSE?"
3000 PRINT
3010 INPUT"ENTER TRUE or FALSE";T$
3020 PRINT
3030 IF T$ = "FALSE" GOTO 3090
3040 PRINT"WRONG - the correct answer is FALSE"
3050 PRINT"       CLEAR statement is for clearing any space more than 50"
3060 PRINT"       and it also initializes numeric variables to 0."
3070 PRINT"       See part 2, CLEAR."
3080 GOTO 3110
3090 PRINT"CORRECT"
3100 Q(10) = 1
3110 PRINT
3120 INPUT"press ENTER";T$
3130 GOSUB 4080
3140 FOR X = 1 TO 10
3150    Y = Y+Q(X)
3160 NEXT X
3170 PRINT"You have finished the test, out of 10 possible correct answers"
3180 PRINT"you scored "Y"."
3190 PRINT
3200 IF Y > 6 THEN  PRINT"YOU HAVE PASSED"
3210 GOSUB 3930
3220 IF Y > 6 THEN GOTO 3300
3230 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
3240 PRINT
3250 PRINT"YOU SHOULD RETAKE LESSON 3!"
3260 PRINT
3270 PRINT"You will be returned to the Menu."
3280 PRINT
3290 GOTO 4120
```

```
3300 PRINT
3310 PRINT"Do you want your score recorded on a permanent file?"
3320 PRINT
3330 PRINT"A  YES"
3340 PRINT"B  NO"
3350 PRINT
3360 INPUT"Which";T$
3370 IF T$ = "B" THEN GOTO 3670
3380 GOSUB 4080
3390 PRINT"To record your score, we must open a file and put your name"
3400 PRINT"in it.  Therefore, surprisingly, we need your name.  If your"
3410 PRINT"name is not unique among the students likely to take this test,"
3420 PRINT"please contact your test monitor for an identifying word that"
3430 PRINT"will make you unique.  Then enter that word below."
3440 PRINT
3450 PRINT"If you have already entered a score previously, be sure to"
3460 PRINT"enter the same name you used before.  (use all capitals)"
3470 PRINT
3480 INPUT"ENTER your word or name now";T$
3490 OPEN"I",1,"SCORES"
3500 X = 0
3510 X = X+1
3520 IF EOF 1 THEN GOTO 3570
3530 INPUT#1,N$(X)
3540 INPUT#1,S(X)
3550 IF N$(X) = T$ THEN GOTO 3750
3560 GOTO 3510
3570 X = X-1
3580 CLOSE
3590 X = X+1
3600 N$(X) = T$
3610 S(X) = Y
3620 OPEN"O",1,"SCORES"
3630 FOR W = 1 TO X
3640   PRINT#1,N$(W)
3650   PRINT#1,S(W)
3660 NEXT W
3670 PRINT
3680 PRINT"You are now qualified to go to LESSON 4."
3690 PRINT
3700 PRINT"If you want a homework assignment, select it now."
3710 PRINT
3720 INPUT"Do you want to see your homework (Y/N)";T$
3730 IF T$ = "N" THEN GOTO 4140
3740 GOTO 4150
3750 S(X) = Y
```

```
3750 X = X+1
3770 IF EOF(1) THEN CLOSE:X=X-1:GOTO 3800
3790 INPUT#1, N$(X), S(X)
3790 GOTO 3760
3800 OPEN"O",1,"SCORE3"
3810    FOR W = 1 TO X
3820        PRINT#1,N$(W)
3830        PRINT#1,S(W)
3840    NEXT W
3850 PRINT
3860 PRINT"You may now take LESSON 4.  You will be returned to the MENU"
3870 PRINT"from where you may go to LESSON 4 or quit."
3880 GOTO 4120
3890 REM **
3900 REM ** THIS ROUTINE LETS STUDENT KNOW WHAT AREA HE
3910 REM ** OR SHE SHOULD REVIEW BEFORE GOING ON
3920 REM **
3930 IF Y=10 THEN RETURN
3940 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
3950 PRINT
3960 IF Q(1) = 0 THEN PRINT "    part 1, KILL statement"
3970 IF Q(2) = 0 OR Q(3) = 0 OR Q(4) = 0 THEN PRINT"    part 1, LOOPs"
3980 IF Q(5) = 0 OR Q(6) = 0 OR Q(7) = 0 OR Q(8) = 0 THEN PRINT"    part 2, ARRAYs"
3990 IF Q(9) = 0 THEN PRINT"    part 2, DIM statement"
4000 IF Q(10) = 0 THEN PRINT"    part 2, CLEAR statement"
4010 PRINT
4020 INPUT"press ENTER";T$
4030 GOSUB 4080
4040 RETURN
4050 REM **
4060 REM ** this subroutine clears the screen
4070 REM **
4080 FOR X = 1 TO 24
4090     PRINT
4100 NEXT X
4110 RETURN
4120 PRINT
4130 INPUT"press ENTER to return to MENU";T$
4140 RUN"MENU"
4150 PRINT
4160 INPUT"press ENTER to load your homework";T$
4170 RUN "HW3"
```

```
1000 REM ** THIS PROGRAM STARTED ON 4 JUNE 1983
1010 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
1020 REM ** TITLE: LESSON 4A
1030 REM **
1040 REM **
1050 REM **
1060 REM **
1070 GOSUB 7020
1080 PRINT"LESSON:  BASIC 4            VERSION: 1 AUGUST 83
1090 PRINT
1100 PRINT"TIME REQUIRED TO COMPLETE LESSON:  About 1.5 hours"
1110 PRINT
1120 PRINT
1130 PRINT"AUTHOR:  Capt Danny J. Creagan"
1140 PRINT"         Air Force Institute of Technology"
1150 PRINT
1160 PRINT"OBJECTIVE:  To teach the student how to make the computer"
1170 PRINT"            communicate with standard peripheral devices."
1180 PRINT
1190 PRINT
1200 PRINT
1210 PRINT
1220 PRINT
1230 INPUT"press the ENTER key to continue";T$
1240 CLEAR 200
1250 LL = 0 'LL IS USED FOR HUMOR IN OPEN STATEMENT SECTION
1260 GOSUB 7020
1270 GOSUB 7070
1280 PRINT
1290 PRINT"A  I'm taking this part in its entirety."
1300 PRINT"B  I wish to review selected areas."
1310 PRINT"C  I want to go to the second part."
1320 PRINT"D  I want to return to the MENU."
1330 PRINT
1340 INPUT"Press either capital A, B, C, or D  and then press ENTER";T$
1350 IF T$ = "C" THEN GOTO 7480
1360 IF T$ = "D" GOTO 7470
1370 IF T$ = "B" GOTO 1510
1380 IF T$ <>"A" GOTO 1290
1390 GOSUB 1620
1400 GOSUB 1960
1410 GOSUB 2980
1420 GOSUB 4630
1430 GOSUB 5900
1440 GOSUB 7020
1450 PRINT"You are now done with this lesson.  If you wish to continue"
```

```
1460 PRINT"to the test, ENTER a T. If you want to review, ENTER an R."
1470 INPUT"ENTER your choice";T$
1480 IF T$ = "T" THEN GOTO 7510
1490 IF T$ = "R" THEN RUN
1500 GOTO 1440
1510 GOSUB 7020
1520 GOSUB 7070
1530 PRINT
1540 PRINT"Please type in the number beside the area you wish"
1550 PRINT"to review (1 through 4) and then press ENTER - ENTER a 0 to go"
1560 PRINT"to the MENU.  ENTER a 5 to go to the second half."
1570 PRINT
1580 INPUT"What is your choice";N
1590 IF N = 0 GOTO 7470
1600 ON N GOSUB 1620 ,1960 ,2980,4630,7480
1610 GOTO 1510
1620 GOSUB 7020
1630 PRINT"                              Introduction"
1640 PRINT
1650 PRINT"In this lesson we will start learning how to communicate with"
1660 PRINT"our disk drives and printers.  The first section deals with"
1670 PRINT"printing our programs on a printer.  We can either print the"
1680 PRINT"output (ie the answer that our program calculated), or we can"
1690 PRINT"print our program listing to a printer."
1700 PRINT
1710 PRINT"The remaining sections will show us how to store data on a"
1720 PRINT"disk, so that we can save important calculations for future"
1730 PRINT"use."
1740 PRINT
1750 PRINT"Both sections are very important to the programmer and you "
1760 PRINT"will find yourself using them often."
1770 PRINT
1780 INPUT"press ENTER";T$
1790 GOSUB 7020
1800 PRINT"                              Introduction"
1810 PRINT
1820 PRINT"From this lesson to the end of your training, we will be "
1830 PRINT"covering areas that are complex and difficult to remember."
1840 PRINT"Therefore, we recommend that you have your BASIC manual with"
1850 PRINT"you at all times.  When we ask you a question, and you are not"
1860 PRINT"sure about the answer, LOOK IT UP IN YOUR MANUAL.  The answer"
1870 PRINT"will also be in the lesson, but you should get used to using"
1880 PRINT"the manual.  You cannot memorize all the rules in a few weeks"
1890 PRINT"or months.  So be sure to keep your reference book handy,"
1900 PRINT"whether you are taking a test, or making your own program."
1910 PRINT
```

```
1920 PRINT"Remember, looking up the answers is not cheating, its LEARNING."
1930 PRINT
1940 INPUT"press ENTER";T$
1950 RETURN
1960 GOSUB 7020
1970 PRINT"                        LPRINT & LLIST"
1980 PRINT
1990 PRINT"LPRINT and LLIST allow you to output information to the line"
2000 PRINT"printer. They are extremely simple to use and they work "
2010 PRINT"almost exactly like the PRINT and LIST words.  There are"
2020 PRINT"only a couple of things you need to keep in mind when you"
2030 PRINT"use them."
2040 PRINT
2050 PRINT"First, make sure the printer is hooked up and turned on."
2060 PRINT"       and, if you are using a Cromemco, make sure the "
2070 PRINT"       printer is LINKed to your terminal. (ask your"
2080 PRINT"       operator how to use the LINK command."
2090 PRINT
2100 PRINT"Second, make sure there is enough paper in the printer."
2110 PRINT"      to do your whole job."
2120 PRINT
2130 INPUT"press ENTER";T$
2140 GOSUB 7020
2150 PRINT"                        LPRINT & LLIST"
2160 PRINT
2170 PRINT"To LPRINT a STRING to the printer, you must enclose it in"
2180 PRINT"quotes, just like the PRINT statement.  When you LPRINT a"
2190 PRINT"numerical variable, you do not enclose it in quotes."
2200 PRINT"You will not see either on the screen. They will only print"
2210 PRINT"on the printer.  Examples of valid LPRINT statements are:"
2220 PRINT
2230 PRINT"  10 LPRINT "CHR$(34)"MONTHLY TRAINING REPORT"CHR$(34)
2240 PRINT
2250 PRINT"  10 A = 10
2260 PRINT"  20 LPRINT A"
2270 PRINT
2280 INPUT"press ENTER";T$
2290 GOSUB 7020
2300 PRINT"                        LPRINT & LLIST"
2310 PRINT
2320 PRINT"You may LPRINT TABs also.  (just as you can PRINT TABs)"
2330 PRINT"However, the TAB function reacts differently on different"
2340 PRINT"machines.  The general format for LPRINTing a TAB is:"
2350 PRINT
2360 PRINT"   10  LPRINT TAB(X)"CHR$(34)"data"CHR$(34)
2370 PRINT
```

```
2380 PRINT"Where X is a number between 0 and the length of your printer"
2390 PRINT"carriage.  When used, the carriage will go over X columns"
2400 PRINT"before it starts to print.  Some computers will not TAB past"
2410 PRINT"80 columns.  You may wish to experiment with yours to see"
2420 PRINT"what its limitations are.  Note that there isn't a space be-"
2430 PRINT"tween the TAB command and the data."
2440 PRINT
2450 INPUT"press ENTER";T$
2460 GOSUB 7020
2470 PRINT"                         LPRINT & LLIST"
2480 PRINT
2490 PRINT"LLIST works just like LIST only it outputs to the printer"
2500 PRINT
2510 PRINT"It is normally used from the IMMEDIATE mode when you want"
2520 PRINT"to see your program lines on paper (it is sometimes easier"
2530 PRINT"to find 'bugs' in your program if you can see it on a piece"
2540 PRINT"of paper)."
2550 PRINT
2560 PRINT"LLIST, and LLIST 100-400 are valid LLIST commands.  LLIST"
2570 PRINT"100-400 lists lines 100-400 to the printer."
2580 PRINT
2590 PRINT
2600 INPUT"press ENTER";T$
2610 GOSUB 7020
2620 PRINT"LPRINT and LLIST output data to the printer and to"
2630 PRINT"the screen."
2640 PRINT
2650 PRINT"Is the above sentence TRUE or FALSE?"
2660 PRINT
2670 PRINT"A  TRUE"
2680 PRINT"B  FALSE"
2690 PRINT
2700 INPUT"ENTER the letter opposite the correct answer and press ENTER";T$
2710 PRINT
2720 IF T$ = "B" THEN GOTO 2750
2730 PRINT"WRONG - LLPRINT & LLIST only output to the printer"
2740 GOTO 2760
2750 PRINT"CORRECT"
2760 PRINT
2770 INPUT"press ENTER";T$
2780 GOSUB 7020
2790 PRINT"Which of the following commands will LLIST your entire"
2800 PRINT"program to the printer?"
2810 PRINT
2820 PRINT"A  LLIST"
2830 PRINT"B  LLIST ALL"
```

```
2840 PRINT"C  LLIST "CHR$(34)"ALL"CHR$(34)
2850 PRINT"D  LLIST "CHR$(34)"lfn"CHR$(34)"  (lfn is your program name)"
2860 PRINT
2870 INPUT"ENTER the letter opposite the correct answer";T$
2880 PRINT
2890 IF T$="A" THEN GOTO 2920
2900 PRINT"WRONG - LLIST is just like LIST, correct answer is A"
2910 GOTO 2930
2920 PRINT"CORRECT"
2930 PRINT
2940 INPUT"press ENTER";T$
2950 GOSUB 7180
2960 IF T$ = "B" GOTO 1960
2970 RETURN
2980 GOSUB 7020
2990 PRINT"                    Sequential Files Intro"
3000 PRINT
3010 PRINT"Remember, in the previous lessons, we learned that we could"
3020 PRINT"INPUT data into our programs, but that the data was not stored"
3030 PRINT"permanently.  If we turned the machine off or otherwise ended"
3040 PRINT"our program, all the data that we ENTERed was lost.  If we"
3050 PRINT"wanted to RUN the program again, we had to re-ENTER the data."
3060 PRINT
3070 PRINT"In the rest of this lesson, we will learn how to store our data"
3080 PRINT"that we ENTERed on a disk FILE.  When we do that, we can always"
3090 PRINT"recall it for future use, and we won't have to keep entering"
3100 PRINT"the same information everytime we run a program.  We just have"
3110 PRINT"to tell the program to read the data from a disk.  Our life"
3120 PRINT"with our computer then becomes much easier."
3130 PRINT
3140 INPUT"press ENTER";T$
3150 GOSUB 7020
3160 PRINT"                           Sequential Files"
3170 PRINT
3180 PRINT"A disk file is an organized collection of data, such as a "
3190 PRINT"training record, or a mailing list.  It is usually composed of"
3200 PRINT"JUST the data, and nothing else.  Program statements or BASIC"
3210 PRINT"words are normally not stored in file format."
3220 PRINT
3230 INPUT"press ENTER";T$
3240 GOSUB 7020
3250 PRINT"                           Sequential Files "
3260 PRINT
3270 PRINT
3280 PRINT"To transfer data from a BASIC program to a disk file, you must"
3290 PRINT"create a BUFFER in memory.  The data is first transferred to "
```

```
3300 PRINT"the BUFFER, and then it is processed and written to the disk."
3310 PRINT"There are two types of files that we can use in BASIC.  They"
3320 PRINT"are SEQUENTIAL files and DIRECT access files.  We will only"
3330 PRINT"explain SEQUENTIAL files in this lesson.  If you need to "
3340 PRINT"learn DIRECT access files, this lesson will still help you"
3350 PRINT"because many of the commands are similar.  Your BASIC manual"
3360 PRINT"will explain the differences (DIRECT access is often called"
3370 PRINT"RANDOM access by some manuals)."
3380 PRINT
3390 INPUT"press ENTER";T$
3400 GOSUB 7020
3410 PRINT"Is the following statement TRUE or FALSE?"
3420 PRINT
3430 PRINT"Sequential files do not need a BUFFER in memory, but DIRECT"
3440 PRINT"access files do."
3450 PRINT
3460 PRINT"A  TRUE"
3470 PRINT"B  FALSE"
3480 PRINT"C  I DON'T KNOW"
3490 PRINT
3500 INPUT"ENTER the letter opposite the correct answer";T$
3510 PRINT
3520 IF T$ = "C" THEN GOSUB 7020:PRINT"Make a guess":PRINT:GOTO 3430
3530 IF T$ = "B" THEN GOTO 3570
3540 PRINT"WRONG - you ALWAYs have to create a BUFFER in memory"
3550 PRINT"        we will show you how in the next parts of the lesson."
3560 GOTO 3590
3570 PRINT"CORRECT - we will show you how to create the BUFFER in the"
3580 PRINT"          next parts of the lesson."
3590 PRINT
3600 INPUT"press ENTER";T$
3610 GOSUB 7020
3620 PRINT"                    Sequential Files "
3630 PRINT
3640 PRINT"With a sequential file, you must access the data the same way"
3650 PRINT"you wrote it to the disk.  If you were to write the ages of "
3660 PRINT"three people to the disk, using sequential access mode, you"
3670 PRINT"could only read the last age you entered by reading ages one"
3680 PRINT"and two first.  They would be stored in a line, and the"
3690 PRINT"computer would have to start with the first age and search"
3700 PRINT"through the list in order, until it found the last age."
3710 PRINT"Even though this is not a fast way of handling files, the"
3720 PRINT"computer still does a good job of it.  Your scores for the"
3730 PRINT"previous tests were logged into a file using SEQUENTIAL access"
3740 PRINT"mode."
3750 PRINT
```

```
3760 INPUT"press ENTER";T$
3770 GOSUB 7020
3780 PRINT"If you wrote two names to a disk file using SEQUENTIAL access"
3790 PRINT"mode, could you get to the LAST name you wrote without reading"
3800 PRINT"the FIRST name you wrote?"
3810 PRINT
3820 PRINT"A  YES"
3830 PRINT"B  NO"
3840 PRINT"C  I wish to take the 5th"
3850 PRINT
3860 INPUT"ENTER the letter opposite the correct answer";T$
3870 PRINT
3880 IF T$ = "C" THEN GOSUB 7020: PRINT"Have you always had a drinking problem?":PRINT:PRINT"Try
     again":PRINT:GOTO 3780
3890 IF T$ = "B" THEN GOTO 3920
3900 PRINT"WRONG - the correct answer is B "
3910 GOTO 3930
3920 PRINT"CORRECT - good answer!"
3930 PRINT
3940 INPUT"press ENTER";T$
3950 GOSUB 7020
3960 PRINT"                    Sequential Files Intro"
3970 PRINT
3980 PRINT"The statements and functions used with sequential files are:"
3990 PRINT
4000 PRINT"     OPEN           PRINT#         EOF"
4010 PRINT"     CLOSE          INPUT#         CLOSE"
4020 PRINT
4030 PRINT"We will be covering these words in greater detail in the"
4040 PRINT"following sections.  You don't have to memorize them now, just"
4050 PRINT"look them over. Generally, you would OPEN the file, either"
4060 PRINT"INPUT# your data, or PRINT# it out to the disk, and then you"
4070 PRINT"would CLOSE the file before going on."
4080 PRINT
4090 INPUT"press ENTER";T$
4100 GOSUB 7020
4110 GOSUB 7290
4120 PRINT"This is an example of how to create a SEQUENTIAL file, write"
4130 PRINT"some data to it, and then CLOSE it.  We will be examining this"
4140 PRINT"program, and a companion program that will INPUT data from"
4150 PRINT"the file we created, in the next sections.  Generally speaking"
4160 PRINT"line 10 OPENs the file (we'll explain more later), line 20"
4170 PRINT"asks the operator to INPUT his/her name, and line 30 writes"
4180 PRINT"the name out to a file called 'TEST'.  Line 40 CLOSES the file."
4190 PRINT
4200 INPUT"press ENTER";T$
```

```
4210 GOSUB 7020
4220 PRINT"Sequential access files are written to a disk in order, and"
4230 PRINT"you cannot access a piece of information from the middle of the"
4240 PRINT"file without searching through all the records for data pieces,"
4250 PRINT"from first to the one that has the record you want."
4260 PRINT
4270 PRINT"Is the above statement TRUE or FALSE?"
4280 PRINT
4290 PRINT"A  TRUE"
4300 PRINT"B  FALSE"
4310 PRINT"C  Uhhh... somewhere in-between?"
4320 PRINT
4330 INPUT"ENTER the letter opposite the correct answer";T$
4340 PRINT
4350 IF T$ = "A" GOTO 4410
4360 IF T$ = "C" THEN PRINT"Uhhhh...";
4370 PRINT"WRONG - This is a key concept, please go back and"
4380 PRINT"        review this section before going on.  You will be"
4390 PRINT"        given the opportunity to review in a few moments"
4400 GOTO 4420
4410 PRINT"CORRECT - THAT WAS IMPORTANT TO UNDERSTAND!"
4420 PRINT
4430 INPUT"press ENTER";T$
4440 GOSUB 7020
4450 PRINT"The two types of file modes, SEQUENTIAL and DIRECT."
4460 PRINT
4470 PRINT"Is the above statement TRUE or FALSE?"
4480 PRINT
4490 PRINT"A  TRUE"
4500 PRINT"B  FALSE"
4510 PRINT
4520 INPUT"ENTER the letter opposite the correct answer";T$
4530 PRINT
4540 IF T$ = "A" GOTO 4570
4550 PRINT"WRONG - the two modes ARE called SEQUENTIAL and DIRECT"
4560 GOTO 4580
4570 PRINT"CORRECT !"
4580 PRINT
4590 INPUT"press ENTER";T$
4600 GOSUB 7180
4610 IF T$ = "B" GOTO 2980
4620 RETURN
4630 GOSUB 7020
4640 PRINT"                              OPEN statement"
4650 PRINT
4660 GOSUB 7230
```

```
4670 PRINT"In the above program, line 10 OPENs the file we wish to make."
4680 PRINT"Whenever you work with a file, you MUST OPEN it first, then"
4690 PRINT"manipulate the data, and then CLOSE it.  If you try to write"
4700 PRINT"DATA to a disk without OPENing the file, you will get an ERROR"
4710 PRINT"message.   (also, don't OPEN a file that is already OPEN)"
4720 PRINT
4730 PRINT
4740 INPUT"press ENTER";T$
4750 GOSUB 7020
4760 PRINT"Can you OPEN a file that is already OPEN?"
4770 PRINT
4780 PRINT"A  YES"
4790 PRINT"B  NO"
4800 PRINT
4810 INPUT"ENTER the letter opposite the correct answer";T$
4820 PRINT
4830 IF T$ ="B" THEN GOTO 4860
4840 PRINT"WRONG - the correct answer is B"
4850 GOTO 4870
4860 PRINT"CORRECT"
4870 PRINT
4880 INPUT"press ENTER";T$
4890 GOSUB 7020
4900 GOSUB 7280
4910 PRINT"Line 10 OPENS the file in this manner:"
4920 PRINT
4930 PRINT"OPEN is the keyword that signals the computer to expect three"
4940 PRINT"more pieces of information.  The first piece is the letter 'O'"
4950 PRINT"or 'I'.  'O' stands for OUTPUT, and I stands for INPUT.  The'
4960 PRINT"next bit of data is the BUFFER number.  In Microsoft BASIC you"
4970 PRINT"may have up to 8 buffers (more on some versions).  For our pur-"
4980 PRINT"poses, we will use buffer #1.  The last bit of data is the"
4990 PRINT"filename.  Notice that the file mode and filename are in quotes"
4999 PRINT
5000 INPUT"press ENTER";T$
5010 GOSUB 7020
5020 GOSUB 7280
5030 PRINT"To recap then, the format for the OPEN statement is:"
5040 PRINT
5050 PRINT"      OPEN "CHR$(34)"(mode)"CHR$(34)",buffer #,"CHR$(34)"(filename)"CHR$(34)
5060 PRINT
5070 PRINT"If you are OUTPUTing DATA the mode is 'O', if you are"
5080 PRINT"INPUTing data, the mode is 'I'.  You can have up to 3 buffers"
5090 PRINT"(you must declare any buffers over 3 when working with the TRS-"
5100 PRINT"80  just answer the BASIC startup dialog with the correct "
5110 PRINT"number of files - 3 is the default).  The filename must be in"
```

```
5120 PRINT"quotes."
5129 PRINT
5130 INPUT"press ENTER";T$
5140 GOSUB 7020
5150 PRINT"What are the two file modes?"
5160 PRINT
5170 PRINT"A  INPUT and OUTPUT"
5180 PRINT"B  "CHR$(34)"I"CHR$(34)" and "CHR$(34)"O"CHR$(34)
5190 PRINT"C  1 through 4 and 4 through 8"
5200 PRINT"D  None of the above"
5210 PRINT
5220 INPUT"ENTER the letter opposite the correct answer";T$
5230 PRINT
5240 IF T$ = "B" THEN GOTO 5270
5250 PRINT"WRONG - the correct answer is B"
5260 GOTO 5280
5270 PRINT"CORRECT - Way to go!"
5280 PRINT
5290 INPUT"press ENTER";T$
5300 GOSUB 7020
5310 PRINT"                              OPEN statement"
5320 PRINT
5330 GOSUB 7350
5340 PRINT"Here is an example of an OPEN statement (line 10) that opens"
5350 PRINT"a file for INPUT.  Notice that the mode is 'I'."
5360 PRINT
5370 PRINT"Also notice that there is NOT a comma between the OPEN word"
5380 PRINT"and the MODE, but that all the rest of the terms are separated"
5390 PRINT"by a comma."
5400 PRINT
5410 INPUT"press ENTER";T$
5420 GOSUB 7020
5430 PRINT"What is the significance of file modes 'O' and 'I'?"
5440 PRINT
5450 PRINT"A  'O' tells the computer that you are going to write to disk"
5460 PRINT"   and 'I' tells the computer you are going to input from disk"
5470 PRINT
5480 PRINT"B  'O' tells the computer that the files section is ON and"
5490 PRINT"   'I' tells the computer that you want to INTERROGATE"
5500 PRINT
5510 PRINT"C  'O' tells the computer you want to Organize files and 'I'"
5520 PRINT"   tells NASA to launch the shuttle."
5530 PRINT
5540 INPUT"ENTER the letter opposite the correct answer";T$
5550 PRINT
5560 IF T$ = "A" THEN GOTO 5600
```

```
5570 IF T$ = "C" THEN GOSUB 7020:LL = 1:PRINT"Starting countdown now.....":PRINT:INPUT"press ENTER for
     LAUNCH back to the question";T$:GOTO 5420
5580 PRINT"WRONG - the correct answer is A"
5590 GOTO 5610
5600 PRINT"CORRECT"
5610 PRINT
5620 INPUT"press ENTER";T$
5630 IF LL = 1 THEN PRINT:PRINT"FIVE HOURS TO SHUTTLE LAUNCH .... COUNTDOWN CONTINUES":PRINT:INPUT"PRESS
     ENTER";T$
5640 GOSUB 7020
5650 PRINT"                              OPEN Statement"
5660 PRINT
5670 GOSUB 7290
5680 PRINT"The buffer number can be any number between 1 and 3.  If we"
5690 PRINT"use the buffer for one file, and later we OPEN another file in"
5700 PRINT"the same program, we cannot use the same buffer number.  It"
5710 PRINT"MUST BE DIFFERENT.  If there is more than one file OPEN at"
5720 PRINT"the same time, then they must be using different buffers!!"
5730 PRINT
5740 INPUT"press ENTER";T$
5750 IF LL = 1 THEN PRINT:PRINT"FOUR HOURS TO SHUTTLE LAUNCH .... COUNTDOWN CONTINUES":PRINT:INPUT"PRESS
     ENTER";T$
5760 GOSUB 7020
5770 PRINT"How many files with the same buffer number can we have OPEN"
5780 PRINT"at the same time?"
5790 PRINT
5800 PRINT"A  ONE          B  TWO          C THREE       D  FOUR"
5810 PRINT
5820 INPUT"ENTER the letter opposite the correct answer";T$
5830 PRINT
5840 IF T$="A" THEN GOTO 5870
5850 PRINT"WRONG ! the correct answer is A"
5860 GOTO 5880
5870 PRINT"CORRECT"
5880 PRINT
5890 INPUT"press ENTER";T$
5900 GOSUB 7020
5910 GOSUB 7280
5920 PRINT"You must enclose the filename in quotes if it is a character"
5930 PRINT"string.  However, the following is also legal:"
5940 GOSUB 7420
5950 INPUT"press ENTER";T$
5960 GOSUB 7020
5970 GOSUB 7420
5980 PRINT"Note that a STRING variable may take the place of the filename"
5990 PRINT"AS LONG AS THE STRING VARIABLE IS SET EQUAL TO A VALID NAME !!"
```

```
6000 PRINT
6010 PRINT"If you use a STRING variable for a filename, you do not enclose"
6020 PRINT"it in quotes.  Look at the example above."
6030 PRINT
6040 INPUT"press ENTER";T$
6050 GOSUB 7020
6060 PRINT"Is the following statement valid?"
6070 PRINT
6080 PRINT"10 OPEN"CHR$(34)"I"CHR$(34)",2,"CHR$(34)"XX$"CHR$(34)
6090 PRINT
6100 PRINT"A  YES"
6110 PRINT"B  NO"
6120 PRINT
6130 INPUT"ENTER the letter opposite the correct answer";T$
6140 PRINT
6150 IF T$ = "B" THEN GOTO 6180
6160 PRINT"WRONG - the correct answer is B. XX$ should not have quotes"
6170 GOTO 6190
6180 PRINT"CORRECT"
6190 PRINT
6200 INPUT"press ENTER";T$
6210 IF LL = 1 THEN PRINT: PRINT"THREE HOURS TO LAUNCH .... COUNTDOWN CONTINUES":PRINT:INPUT"PRESS ENTER";T$
6220 GOSUB 7020
6230 GOSUB 7280
6240 PRINT"When line 10 OPENs the file 'TEST', the computer searches"
6250 PRINT"available disk space to see if the file already exists. IF IT"
6260 PRINT"DOESN'T EXIST, THE COMPUTER WILL CREATE IT AUTOMATICALLY!!!"
6270 PRINT"(this is only true for the OUTPUT mode. if INPUT mode, the file"
6280 PRINT"must have existed previously or BASIC will print an ERROR)"
6290 PRINT
6300 PRINT"REMEMBER ALSO, unless you've CLOSED a file that has previously
6310 PRINT"been opened, you cannot use the buffer number again !"
6320 PRINT
6330 INPUT"press ENTER";T$
6340 GOSUB 7020
6350 GOSUB 7280
6360 PRINT"If the file 'TEST' exists, then line 10 will re-open it. line"
6370 PRINT"30 will WRITE OVER THE PREVIOUS DATA IN THE FILE, and line"
6380 PRINT"40 will CLOSE the file.  EVEN IF THE FILE WAS SEVERAL THOUSAND"
6390 PRINT"WORDS LONG, AFTER THE ABOVE PROGRAM IS RUN, IT WILL ONLY BE"
6400 PRINT"AS LONG AS THE NAME THAT WAS ENTERED IN LINE 20!  "
6410 PRINT
6420 PRINT"Sequential files must be loaded into memory, manipulated and"
6430 PRINT"then written back out in their entirety, you cannot just write"
6440 PRINT"a single record onto the front of the file that's on the disk."
6450 PRINT
```

```
6460 INPUT"press ENTER";T$
6470 GOSUB 7020
6480 PRINT"Write in the statement that will OPEN a file for OUTPUT to disk"
6490 PRINT"and use buffer number 3.  The filename is TWIG"
6500 PRINT
6510 PRINT"Do not use a line number (although you would normally), and do"
6520 PRINT"NOT put in ANY blanks."
6530 PRINT
6540 LINE INPUT"ENTER your answer ? ";T$
6550 PRINT
6560 G$="OPEN"+CHR$(34)+"O"+CHR$(34)+",3,"+CHR$(34)+"TWIG"+CHR$ 34
6570 Y$ ="OPEN "+CHR$(34)+"O"+CHR$(34)+",3,"+CHR$.34)+"TWIG"+CHR$(34
6580 IF T$=G$ OR T$ = Y$  THEN GOTO 6630
6590 PRINT"WRONG - this section is difficult to visualize sometimes. how-"
6600 PRINT"ever, it is very important. You may wish to review it after"
6610 PRINT"this test.  The correct answer is: "G$
6620 GOTO 6640
6630 PRINT"Fabulous  Without a doubt you are a programmer  "
6640 PRINT
6650 INPUT"press ENTER";T$
6660 GOSUB 7020
6670 PRINT"When a sequential file is OPENed that formerly had your "
6680 PRINT"training records in it, and you didn't want the DATA destroyed."
6690 PRINT"what must you be sure to do?"
6700 PRINT
6710 PRINT"A  Not write in the middle of the file unless it is mode "O""
6720 PRINT"B  INPUT# all the data, manipulate it, then PRINT# it back out"
6730 PRINT"C  Nothing, you cannot OPEN a file that was previously created
6740 PRINT"D  Write only on the END of the file"
6750 PRINT
6760 INPUT"ENTER the letter opposite the correct answer ";T$
6770 PRINT
6780 IF T$="B" THEN GOTO 6810
6790 PRINT"WRONG - the correct answer is B"
6800 GOTO 6820
6810 PRINT"CORRECT"
6820 PRINT
6830 INPUT"press ENTER";T$
6840 IF LL = 1 THEN PRINT:PRINT"SHUTTLE LAUNCH ABORTED .... COMPUTER MALFUNCTION BLAMED":PRINT:INPUT"PRESS
     ENTER";T$
6850 GOSUB 7130
6860 IF T$="S" GOTO 4630
6870 RETURN
6880 GOSUB 7120
6890 RETURN
6900 GOSUB 7020
```

```
6910 PRINT"You have completed this portion of lesson 4.  If you wish "
6920 PRINT"to continue on to the next half of the lesson, enter a "C"."
6930 PRINT"If you wish to review this lesson again, enter an "R"."
6940 PRINT
6950 INPUT"Enter your choice now";T$
6960 IF T$ ="R" THEN RUN
6970 IF T$ <> "C" THEN GOTO 6900
6980 GOTO 7480
6990 REM **
7000 REM ** clear screen subroutine
7010 REM **
7020 FOR X = 1 TO 24
7030    PRINT
7040 NEXT X
7050 RETURN
7060 REM ** This is the menu subroutine
7070 PRINT"                    LESSON 4"
7080 PRINT
7090 PRINT"This is the first part of a two part lesson.  It is divided"
7100 PRINT"into the following sections:"
7110 PRINT
7120 PRINT"1)  Introduction            3)  Sequential Files Intro."
7130 PRINT"2)  LPRINT & LLIST          4)  OPENing files"
7140 PRINT
7150 PRINT
7160 RETURN
7170 REM **  This subroutine lets student review a section
7180 GOSUB 7020
7190 PRINT"Which do you wish to do?"
7200 PRINT
7210 PRINT"A  Continue on"
7220 PRINT"B  Review this section again"
7230 PRINT
7240 INPUT"press the letter opposite the correct answer and press ENTER";T$
7250 IF T$ = "A" OR T$ = "B" THEN RETURN
7260 GOTO 7190
7270 REM ** this subroutine is for sequential intro. example
7280 PRINT"10 OPEN "CHR$(34)"O"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
7290 PRINT"20 INPUT"CHR$(34)"Type in your name and press ENTER"CHR$(34)";T$"
7300 PRINT"30 PRINT#1,T$"
7310 PRINT"40 CLOSE 1"
7320 PRINT
7330 RETURN
7340 REM ** This subroutine is for the sequential files intro example
7350 PRINT"10 OPEN "CHR$(34)"I"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
7360 PRINT"20 INPUT#1,N$"
```

```
7370 PRINT"30 PRINT N$"
7380 PRINT"40 CLOSE 1
7390 PRINT
7400 RETURN
7410 REM ** This subroutine is for the OPEN statements example
7420 PRINT
7430 PRINT"10 A$ = "CHR$(34)"TEST"CHR$(34)
7440 PRINT"20 OPEN "CHR$(34)"O"CHR$(34)",1,A$"
7450 PRINT
7460 RETURN
7470 RUN "MENU"
7480 PRINT
7490 PRINT"Going to the second half of the lesson, wait one moment"
7500 RUN "LESSON4A"
7510 GOSUB 7020
7520 PRINT"Going to TEST 4 - please standby."
7530 RUN "TEST4"
7540 END
```

```
1000 CLEAR 200
1010 GOSUB 5200
1020 GOSUB 5250
1030 PRINT
1040 PRINT"A  I'm taking this part in its entirety."
1050 PRINT"B  I wish to review selected areas (or take the test)."
1060 PRINT"C  I want to return to the Menu."
1070 PRINT"D  I want to go the the first part."
1080 PRINT
1090 INPUT"Press either capital A, B, C or D  and then press ENTER";T$
1100 IF T$ = "C" GOTO 5740
1110 IF T$ = "D" THEN GOTO 5770
1120 IF T$ = "B" GOTO 1250
1130 IF T$ <>"A" GOTO 1040
1140 GOSUB 1360
1150 GOSUB 1770
1160 GOSUB 2580
1170 GOSUB 4650
1180 GOSUB 5200
1190 PRINT"You are now done with this lesson.  If you wish to continue"
1200 PRINT"to the test, ENTER a T. If you want to review, ENTER an R."
1210 INPUT"ENTER your choice";T$
1220 IF T$ = "T" THEN GOTO 5730
1230 IF T$ = "R" THEN RUN
1240 GOTO 1190
1250 GOSUB 5200
1260 GOSUB 5250
1270 PRINT
1280 PRINT"Please type in the number beside the area you wish"
1290 PRINT"to review (1 through 5) and then press ENTER - ENTER a 0 to go"
1300 PRINT"to the MENU."
1310 PRINT
1320 INPUT"What is your choice";N
1330 IF N = 0 GOTO 5740
1340 ON N GOSUB 1360 ,1770 ,2580,4650,5730
1350 GOTO 1250
1360 GOSUB 5200
1370 PRINT"                              CLOSE Statement"
1380 PRINT
1390 PRINT"We already stated previously that the CLOSE statement was"
1400 PRINT"necessary after you were done manipulating your files.  It"
1410 PRINT"has a few variations that are nice to know."
1420 PRINT
1430 PRINT"          CLOSE (buffer #.. .. .. ..)
1440 PRINT
1450 PRINT"The format for CLOSE is the BASIC word CLOSE plus an optional"
```

```
1460 PRINT"buffer number.  If you include a buffer number, just that "
1470 PRINT"buffer will be closed.  If you do not include a buffer number"
1480 PRINT"then ALL the buffers that were OPEN will be closed!"
1490 PRINT
1500 INPUT"press ENTER";T$
1510 G$ = "OPEN"+CHR$(34)+"O"+CHR$(34)+",3,"+CHR$(34)+"QUESTION"+CHR$(34)
1520 GOSUB 5200
1530 PRINT"Is the following program valid (lines 20 & 30 are good)"
1540 PRINT
1550 PRINT"10  "G$
1560 PRINT"20  INPUT A$"
1570 PRINT"30  PRINT#3,A$"
1580 PRINT"40  CLOSE"
1590 PRINT"50  "G$
1600 PRINT".....etc"
1610 PRINT
1620 PRINT"A   YES it is valid"
1630 PRINT"B   NO it will fail because line 50 OPENs a file already used"
1640 PRINT
1650 INPUT"ENTER the letter opposite the correct answer";T$
1660 PRINT
1670 IF T$ = "A" THEN GOTO 1710
1680 PRINT"WRONG - file #3 was CLOSEd by line 40, so buffer 3 could be"
1690 PRINT"        used again in line 50."
1700 GOTO 1720
1710 PRINT"CORRECT"
1720 PRINT
1730 INPUT"press ENTER";T$
1740 GOSUB 5360
1750 IF T$ = "B" GOTO 1360
1760 RETURN
1770 GOSUB 5200
1780 PRINT"                              PRINT #"
1790 PRINT
1800 GOSUB 5460
1810 PRINT"In line 20 we PRINT to the file buffer (that we OPENed in line"
1820 PRINT"10) the variable T$.  Notice that the number to the right of "
1830 PRINT"the PRINT# corresponds to the buffer number.  If the buffer"
1840 PRINT"was number 3, then, after we OPENed the file, we would "
1850 PRINT"PRINT#3,T$ in line 20"
1860 PRINT
1870 INPUT"press ENTER";T$
1880 GOSUB 5200
1890 PRINT"                              PRINT #"
1900 PRINT
1910 PRINT"PRINT# prints items to a sequential disk file.  When you first"
```

```
1920 PRINT"OPEN the file, the computer sets a pointer at the beginning"
1930 PRINT"of the file, when you tell the computer to PRINT# something,"
1940 PRINT"it starts writing data to the disk at the place where the"
1950 PRINT"pointer is.  At the end of the PRINT# operation, the pointer"
1960 PRINT"advances, so values are written in sequence.  PRINT# writes"
1970 PRINT"data to the disk almost exactly the way PRINT writes data to"
1980 PRINT"the screen (or LPRINT writes to the printer)."
1990 PRINT
2000 INPUT"press ENTER";T$
2010 GOSUB 5200
2020 PRINT"                              PRINT#"
2030 PRINT
2040 PRINT"Commas and semi-colons react the same way with PRINT# that they"
2050 PRINT"do with PRINT statements.  If you were to write the following"
2060 PRINT"program:"
2070 PRINT
2080 PRINT"10  A=10.3"
2090 PRINT"20  B= 20.2"
2100 PRINT
2110 PRINT"to a disk using PRINT#1,A,B (as opposed to PRINT#1,A;B), then"
2120 PRINT"you would put this on the disk : 10.3           20.2"
2130 PRINT
2140 PRINT"See the extra spaces!  Those are 13 blanks that BASIC writes to"
2150 PRINT"the disk."
2160 PRINT
2170 INPUT"press ENTER";T$
2180 GOSUB 5200
2190 PRINT"PRINT#1,A,B"
2200 PRINT
2210 PRINT"The above command puts this on disk : 10.3          20.2"
2220 PRINT
2230 PRINT"If you use a semicolon, like this, PRINT#1,A;B then you get:"
2240 PRINT
2250 PRINT" 10.3   20.2"
2260 PRINT
2270 PRINT"There are only three spaces between the numbers.  So to save"
2280 PRINT"space on the disk, you may want to write to disk using semi-"
2290 PRINT"colons instead of commas between your variables.  Either way"
2300 PRINT"will work, it's just that the commas cause 10 extra blanks to"
2310 PRINT"be PRINT# 'd to the disk."
2320 PRINT
2330 INPUT"press ENTER";T$
2340 GOSUB 5200
2350 PRINT"What is wrong with the following program?"
2360 PRINT
2370 PRINT"10 OPEN"CHR$(34)"I"CHR$(34)",1,"CHR$(34)"NEWPROG"CHR$(34)
```

```
2380 PRINT"20 PRINT#1,34;22;55"
2390 PRINT"30 CLOSE 1"
2400 PRINT
2410 PRINT"A  The mode is incorrect"
2420 PRINT"B  The file buffer is incorrect"
2430 PRINT"C  You cannot use semicolons between numbers in a PRINT#"
2440 PRINT"D  Nothing"
2450 PRINT
2460 INPUT"Enter the letter opposite the correct answer";T$
2470 PRINT
2480 IF T$ = "A" GOTO 2520
2490 PRINT"WRONG - The correct answer is A  - the mode is incorrect for"
2500 PRINT"       PRINT#ing."
2510 GOTO 2530
2520 PRINT"CORRECT - way to GO!"
2530 PRINT
2540 INPUT"press ENTER";T$
2550 GOSUB 5360
2560 IF T$="B" GOTO 1770
2570 RETURN
2580 GOSUB 5200
2590 PRINT"                          INPUT# & EOF"
2600 PRINT
2610 PRINT"INPUT# is similar to the INPUT word that we learned earlier"
2620 PRINT"only it INPUTs data from a disk that previously had data"
2630 PRINT"printed to it."
2640 PRINT
2650 GOSUB 5530
2660 PRINT"The format for the INPUT# statement is similar to the PRINT#"
2670 PRINT"statement.  It is:  INPUT# (buffer#),(variable1),(var2),(etc)"
2680 PRINT
2690 INPUT"press ENTER";T$
2700 GOSUB 5200
2710 PRINT"                          INPUT#"
2720 PRINT
2730 PRINT"INPUT# inputs data from a sequential disk file and stores the"
2740 PRINT"data in a variable.  INPUT# doesn't care how data was placed on"
2750 PRINT"the disk.  It could have been put there with one PRINT#"
2760 PRINT"or twenty PRINT# statements.  WHAT MATTERS TO INPUT# IS HOW THE"
2770 PRINT"DATA IS TERMINATED ON THE DISK, AND WHAT KIND OF DATA IT IS"
2780 PRINT"INPUTTING."
2790 PRINT
2800 INPUT"press ENTER";T$
2810 GOSUB 5200
2820 PRINT"Does the INPUT# statement check to see how the data was"
2830 PRINT"placed on the disk, or does it check to see how the data is"
```

```
2840 PRINT"terminated?"
2850 PRINT
2860 PRINT"A  It only checks to see how it was terminated - it doesn't"
2870 PRINT"   care how the data got there."
2880 PRINT
2890 PRINT"B  It checks to see how the data was placed, it makes a "
2900 PRINT"   difference how many PRINT# statements were used."
2910 PRINT
2920 PRINT"C  Now THIS question is easy.  Just give me a second and I'll"
2930 PRINT"   think of the answer.  Hmmm, let's see....No, don't tell me.."
2940 PRINT
2950 INPUT"ENTER the letter opposite the correct answer";T$
2960 PRINT
2970 IF T$ = "A" THEN GOTO 3010
2980 IF T$ = "C" THEN PRINT"TIMES UP!! .. because you took so much time, you get the answer "
2990 PRINT"WRONG - the answer is A"
3000 GOTO 3020
3010 PRINT"CORRECT - good job"
3020 PRINT
3030 INPUT"press ENTER";T$
3040 GOSUB 5200
3050 PRINT"                          INPUT# & EOF"
3060 PRINT
3070 PRINT"If we are inputting STRING data (our variable is a"
3080 PRINT"STRING such as INPUT#1,N$), INPUT# starts putting data into"
3090 PRINT"the variable starting with the first NON-SPACE it encounters"
3100 PRINT"in the file, and ending when it encounters a carriage return or"
3110 PRINT"a comma, or EOF marker (more about EOF later)."
3120 PRINT
3130 PRINT"If the variable is numeric, then INPUT# fills the variable with"
3140 PRINT"the first character that is not a space or carriage return,"
3150 PRINT"and stops when it encounters another space, comma, or carriage"
3160 PRINT"return or EOF marker."
3170 PRINT
3180 INPUT"press ENTER";T$
3190 GOSUB 5200
3200 PRINT"                          INPUT# and EOF"
3210 PRINT
3220 PRINT"Here is an important concept to understand about how INPUT#"
3230 PRINT"works when you use STRINGs.  IF YOU PRINT A STRING TO DISK, AND"
3240 PRINT"YOU PRINT A NUMBER WITH IT, IT WILL LOOK LIKE THIS ON DISK:"
3250 PRINT
3260 PRINT"       STRING DATA HERE WITH NUMBER FOLLOWING   "1091
3270 PRINT
3280 PRINT"IF YOU INPUT THIS DATA, YOU MUST INPUT IT USING A STRING VARIABLE."
3290 PRINT"WHEN YOU DO, THE COMPUTER WILL PACK THE STRING WITH ALL THE DATA"
```

```
3300 PRINT"ON THE LINE, (INCLUDING THE NUMBER!) BECAUSE A STRING INPUT DOES"
3310 PRINT"NOT RECOGNIZE SPACES AS TERMINATORS, IT ONLY RECOGNIZES CARRIAGE"
3320 PRINT"RETURNS AND COMMAS!"
3330 PRINT
3340 PRINT"Why is this important? Because you wrote the data to disk with"
3350 PRINT"TWO variables, and you read it back with only ONE. If you tried"
3360 PRINT"to read the number after you read in the string, you would not"
3370 PRINT"find it ! "
3380 PRINT
3390 INPUT"press ENTER";T$
3400 GOSUB 5200
3410 PRINT"The solution to the problem of how to write STRING data and"
3420 PRINT"NUMERIC data to disk, is to separate them with carriage returns"
3430 PRINT"(or ENTERs).  That way you can read them back with INPUT#"
3440 PRINT"statement in the same way you wrote it.  For example, if you"
3450 PRINT"write a string and a number to disk, do it this way:"
3460 PRINT
3470 PRINT"10  ..... program assumes file opened correctly....."
3480 PRINT"20  PRINT#1,"CHR$(34)"PRINT THE STRING WITH ONE LINE and the number with another."CHR$(34)
3490 PRINT"30  PRINT#1, 1001"
3500 PRINT
3510 PRINT"And when you read it back, use two separate statements."
3520 PRINT
3530 PRINT"10  INPUT#1,A$"
3540 PRINT"20  INPUT#1,N"
3550 PRINT
3560 PRINT"This will solve the problem of mixing strings and numbers on disk."
3570 PRINT
3580 INPUT"press ENTER";T$
3590 GOSUB 5200
3600 GOSUB 5530
3610 PRINT"If we were to run this program, and the name that was in the"
3620 PRINT"first record in the file was 'DANNY JOE', line 20 would start"
3630 PRINT"with the first byte of the first record that was not a space"
3640 PRINT"or a carriage return, and load N$ with it, it would continue"
3650 PRINT"loading N$ until it encountered either a carriage return or a"
3660 PRINT"comma.  If the data contained a quoted character string, then"
3670 PRINT"all the data between the quotes would be stuffed into N$ "
3680 PRINT"(unless a comma or carriage return were encountered)."
3690 PRINT
3700 INPUT"press ENTER";T$
3710 GOSUB 5200
3720 GOSUB 5530
3730 PRINT"Eventually, and very quickly, the string would be loaded with"
3740 PRINT"the characters 'DANNY JOE', and line 30 would print them out."
3750 PRINT
```

```
3760 PRINT"But what would happen if there was nothing in the file?"
3770 PRINT"The INPUT# statement in line 20 would encounter the EOF marker"
3780 PRINT"and an ERROR would be returned saying that the computer tried"
3790 PRINT"to input data that wasn't there."
3800 PRINT
3810 PRINT"There is a way to test to see if the file is at the end or if"
3820 INPUT"it is empty.     Press ENTER to see what it is";T$
3830 GOSUB 5200
3840 PRINT"The very first thing a computer does when it OPENs a file, is"
3850 PRINT"it puts a marker on the end of it.  The marker is called an EOF"
3860 PRINT"marker.  If we wanted to find out if the end of a file had been"
3870 PRINT"reached, or if the file was empty, we would test it like this:"
3880 PRINT
3890 GOSUB 5650
3900 PRINT
3910 INPUT"press ENTER";T$
3920 GOSUB 5200
3930 GOSUB 5650
3940 PRINT
3950 PRINT"If we had printed a thousand and one names into the file 'TEST'"
3960 PRINT"the computer would put the EOF marker in place # 1002, and"
3970 PRINT"when we used the EOF(buffer #) test after name # 1001, control"
3980 PRINT"would pass to line 60.  All the names would have been printed!"
3990 PRINT"Notice that we never re-OPEN a file if we haven't CLOSED it."
4000 PRINT"If we tried to re-OPEN an already OPEN file, we'ld get an ERROR"
4010 PRINT
4020 INPUT"press ENTER";T$
4030 GOSUB 5200
4040 PRINT"                              INPUT# and EOF"
4050 PRINT
4060 PRINT"The format for the EOF statement is :"
4070 PRINT
4080 PRINT"EOF(buffer #)"
4090 PRINT
4100 PRINT"Where buffer number corresponds to the buffer number of the "
4110 PRINT"file you are testing ."
4120 PRINT
4130 PRINT"Remember to only use the EOF test on a file that is OPEN !"
4140 PRINT
4150 INPUT"press ENTER";T$
4160 GOSUB 5200
4170 PRINT"Suppose we have a file called 'TEST' that is full of numeric"
4180 PRINT"data.  What is wrong with the following program if we were"
4190 PRINT"trying to print the file out to the screen?"
4200 PRINT
4210 PRINT"10 OPEN"CHR$(34)"I"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
```

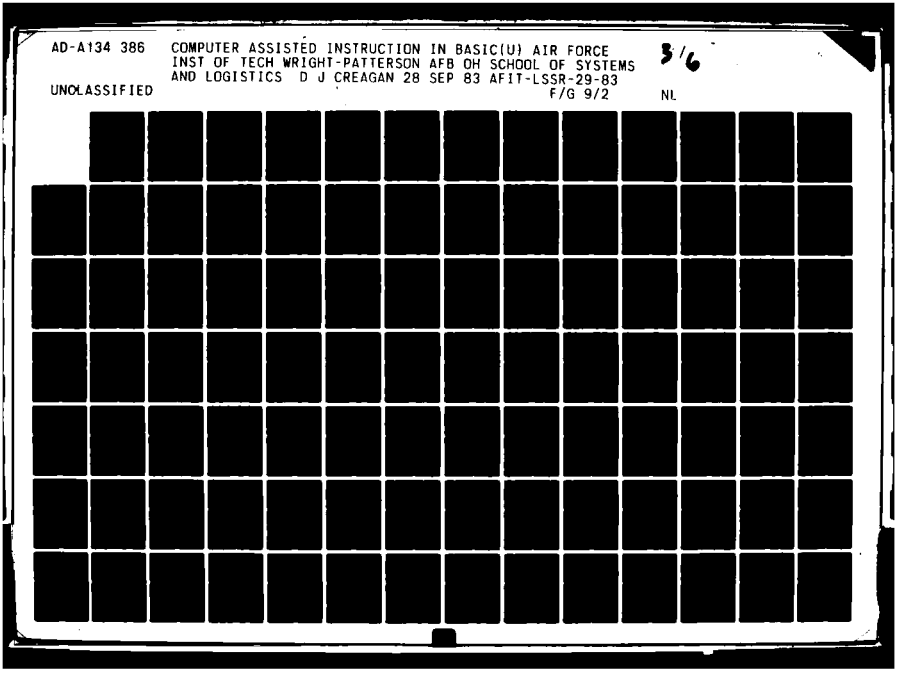
```
4220 PRINT"20 IF EOF(1) THEN STOP"
4230 PRINT"30 INPUT#1,N"
4240 PRINT"40 PRINT N"
4250 PRINT"50 GOTO 10"
4260 PRINT
4270 PRINT"A  Nothing"
4280 PRINT"B  If 'TEST' is empty, the EOF check won't catch it"
4290 PRINT"C  Line 50 should be GOTO 20"
4300 PRINT"D  The file mode is incorrect"
4310 PRINT
4320 INPUT"ENTER the letter opposite the correct answer";T$
4330 PRINT
4340 IF T$ = "C" THEN GOTO 4380
4350 PRINT"WRONG - The problem is that the program trys to re-OPEN the"
4360 PRINT"        file that hasn't been CLOSEd.  Correct answer is C"
4370 GOTO 4390
4380 PRINT"CORRECT"
4390 PRINT
4400 INPUT"press ENTER";T$
4410 GOSUB 5200
4420 PRINT"What is wrong with this program?"
4430 PRINT
4440 PRINT"10 OPEN"CHR$(34)"I"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
4450 PRINT"20 IF EOF(2) THEN STOP"
4460 PRINT"30 INPUT#1,N$"
4470 PRINT"40 PRINT N$"
4480 PRINT"50 GOTO 20"
4490 PRINT
4500 PRINT"A  The wrong buffer number is used"
4510 PRINT"B  Nothing"
4520 PRINT"C  There is no END statement"
4530 PRINT
4540 INPUT"ENTER the letter opposite the correct answer";T$
4550 PRINT
4560 IF T$ = "A" THEN GOTO 4590
4570 PRINT"WRONG the correct answer is A"
4580 GOTO 4600
4590 PRINT"CORRECT"
4600 PRINT
4610 INPUT"press ENTER";T$
4620 GOSUB 5260
4630 IF T$="B" GOTO 2580
4640 RETURN
4650 GOSUB 5200
4660 PRINT"                              SUMMARY"
4670 PRINT
```

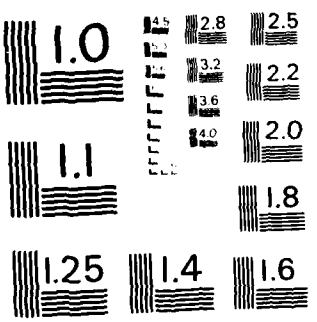
```
4680 PRINT"In this lesson we have learned a great deal about file input"
4690 PRINT"and output.  However, there is a great deal of information"
4700 PRINT"that we have not covered. "
4710 PRINT
4720 PRINT"The purpose of this lesson was to introduce you to the fund-"
4730 PRINT"damental ideas behind sequential files.  You should combine"
4740 PRINT"this knowledge with the previous lessons, and do some outside"
4750 PRINT"studying on your own.  After you take the test, you will be"
4760 PRINT"given an assignment that will include many of the techniques"
4770 PRINT"we have already learned."
4780 PRINT
4790 PRINT"On the following pages, there is a program that uses what we"
4800 PRINT"have learned in this lesson.  Study it carefully."
4810 INPUT"press ENTER";T$
4820 GOSUB 5200
4830 PRINT"The purpose of the program on the following page is to update"
4840 PRINT"a data file that contains STRING data.  It reads in a file"
4850 PRINT"and simultaneously writes out the same data to a different"
4860 PRINT"file.  When you update a file this way, you end up with an"
4870 PRINT"updated file that has a different name than the one you started"
4880 PRINT"with.  While you are studying the program, think about how"
4890 PRINT"you would do it without changing the filename."
4900 PRINT
4910 PRINT"HINT:  the program would have to read the data into an array"
4920 PRINT"and then add data onto the array, and finally write the whole"
4930 PRINT"array back out to the old file."
4940 PRINT
4950 PRINT"Note how the comments are inserted.  Look up why this is legal"
4960 PRINT"in your BASIC manual.  You may be surprised !"
4970 PRINT
4980 INPUT"press ENTER";T$
4990 GOSUB 5200
5000 PRINT"10 CLEAR 500              :'Clears string space"
5010 PRINT"20 INPUT"CHR$(34)"Input filename"CHR$(34)";I$"
5020 PRINT"30 INPUT"CHR$(34)"Output filename"CHR$(34)";O$"
5030 PRINT"40 OPEN"CHR$(34)"I"CHR$(34)",1,I$          :'OPENs the INPUT file"
5040 PRINT"50 OPEN"CHR$(34)"O"CHR$(34)",2,O$          :'OPENs the OUTPUT file"
5050 PRINT"60 IF EOF(1) GOTO 130     :'Checks for end of file in file #1"
5060 PRINT"70 INPUT#1,D$             :'INPUTs data from file #1"
5070 PRINT"80 PRINT#2,D$             :'PRINTs data to the new file"
5080 PRINT"90 GOTO 60                :'Goes back for more from file #1"
5090 PRINT"100 INPUT"CHR$(34)"ENTER DATA (enter 999 to stop)"CHR$(34)";D$"
5100 PRINT"110 IF D$ = "CHR$(34)"999"CHR$(34)" GOTO 140"
5110 PRINT"120 PRINT#2,D$            :'Updates new file with your data"
5120 PRINT"130 GOTO 100              :'Goes back until line 110 sees 999"
5130 PRINT"140 CLOSE"
```

```
5140 PRINT
5150 INPUT"We assume a legal input file previously existed...press ENTER";T$
5160 GOSUB 5360
5170 IF T$="B" GOTO 4650
5180 RETURN
5190 REM ** This subroutine clears the screen on any terminal
5200 FOR I = 1 TO 24
5210    PRINT
5220 NEXT I
5230 RETURN
5240 REM ** This is the menu subroutine
5250 PRINT"                    LESSON 4"
5260 PRINT
5270 PRINT
5280 PRINT"This is the second part of a two part lesson.  It is divided"
5290 PRINT"into the following sections:"
5300 PRINT
5310 PRINT"1)  CLOSE                  3)  INPUT# & EOF"
5320 PRINT"2)  PRINT#                 4)  SUMMARY"
5330 PRINT"                    5)  TEST"
5340 RETURN
5350 REM **  This subroutine lets student review a section
5360 GOSUB 5200
5370 PRINT"Which do you wish to do?"
5380 PRINT
5390 PRINT"A  Continue on"
5400 PRINT"B  Review this section again"
5410 PRINT
5420 INPUT"press the letter opposite the correct answer and press ENTER";T$
5430 IF T$ = "A" OR T$ = "B" THEN RETURN
5440 GOTO 5360
5450 REM ** this subroutine is for sequential intro. example
5460 PRINT"10 OPEN "CHR$(34)"O"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
5470 PRINT"20 INPUT"CHR$(34)"Type in your name and press ENTER"CHR$(34)";T$"
5480 PRINT"30 PRINT#1,T$"
5490 PRINT"40 CLOSE 1"
5500 PRINT
5510 RETURN
5520 REM ** This subroutine is for the sequential files intro example
5530 PRINT"10 OPEN "CHR$(34)"I"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
5540 PRINT"20 INPUT#1,N$"
5550 PRINT"30 PRINT N$"
5560 PRINT"40 CLOSE 1"
5570 PRINT
5580 RETURN
5590 REM ** This subroutine is for the OPEN statements example
```

AD-A134 386 COMPUTER ASSISTED INSTRUCTION IN BASIC(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS D J CREAGAN 28 SEP 83 AFIT-LSSR-29-83 3/6

UNCLASSIFIED F/G 9/2 NL





```
5600 PRINT
5610 PRINT"10 A$ = "CHR$(34)"TEST"CHR$(34)
5620 PRINT"20 OPEN "CHR$(34)"O"CHR$(34)",1,A$"
5630 PRINT
5640 RETURN
5650 PRINT"10 OPEN"CHR$(34)"I"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
5660 PRINT"20 IF EOF(1) THEN GOTO 60"
5670 PRINT"30 INPUT#1,N$"
5680 PRINT"40 PRINT N$"
5690 PRINT"50 GOTO 20"
5700 PRINT"60 PRINT"CHR$(34)"END OF FILE ENCOUNTERED"CHR$(34)
5710 PRINT"70 CLOSE 1"
5720 PRINT".....more of the program or an END statement"
5730 RETURN
5740 RUN "MENU"
5750 PRINT
5760 PRINT"GOING TO THE FIRST PART - WAIT ONE MOMENT"
5770 RUN "LESSON4"
5780 GOSUB 5200"
5790 PRINT"Going to TEST 4 - please standby"
5800 RUN "TEST4"
5810 END
```

```
1000 REM **
1010 REM ** LESSON:  TEST4                VERSION: 1 AUG 83
1020 REM ** AUTHOR:  CAPT DAN CREAGAN
1030 REM **          AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM **            N$(X) = NAMES ARRAY, USED TO READ IN SEQ-
1070 REM **                    UENTIAL NAMES, AND TO WRITE OUT
1080 REM **                    UPDATE NAMES.
1090 REM **            S(X)  = SCORES ARRAY - USED TO READ AND
1100 REM **                    WRITE SCORES
1110 REM **            Q(X)  = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM **                    CORRECT ANSWERS.  IF AN ARRAY
1130 REM **                    ELEMENT EQUALS 1, THE ANSWER WAS
1140 REM **                    CORRECT
1150 REM **
1160 CLEAR 3000
1170 GOSUB 4400
1180 DIM N$(1000)
1190 DIM Q(10)
1200 DIM S(1000)
1210 PRINT"                     FINAL TEST (lesson 4)"
1220 PRINT
1230 PRINT"This test consists of 10 questions, you must get 70 percent"
1240 PRINT"of them correct to pass.  (that's 7 right out of the 10 ques-"
1250 PRINT"tions).  Use only capital letters in your answers, don't"
1260 PRINT"include extra spaces or letters. When you successfully complete"
1270 PRINT"the test, you will be given a homework assignment that will"
1280 PRINT"bring many things together for you.   GOOD LUCK!"
1290 PRINT
1300 INPUT"press ENTER to continue";T$
1310 GOSUB 4400
1320 PRINT"When you LPRINT data you must be sure the printer is on, has"
1330 PRINT"enough paper, and is properly connected.  LPRINTing does not"
1340 PRINT"send output to the screen."
1350 PRINT
1360 PRINT"Is the above paragraph TRUE or FALSE?"
1370 PRINT
1380 PRINT"A  TRUE"
1390 PRINT"B  FALSE"
1400 PRINT
1410 GOSUB 4470
1420 PRINT
1430 IF T$ = "A" THEN GOTO 1470
1440 PRINT"WRONG - the correct answer is A"
1450 PRINT"        part 1, LPRINT"
```

```
1460 GOTO 1490
1470 PRINT"CORRECT"
1480 Q(1) = 1
1490 PRINT
1500 INPUT"press ENTER";T$
1510 GOSUB 4400
1520 PRINT"What is the command to list your program to the printer?"
1530 PRINT
1540 PRINT"A  LPRINT "CHR$(34)"(filename)"CHR$(34)
1550 PRINT"B  LIST "CHR$(34)"(filename)"CHR$(34)
1560 PRINT"C  LLIST"
1570 PRINT"D  None of the above"
1580 PRINT
1590 GOSUB 4470
1600 PRINT
1610 IF T$ = "C" THEN GOTO 1670
1620 PRINT"WRONG - the correct answer is C"
1630 PRINT"        Answer A would print the word 'filename' to the "
1640 PRINT"        printer, answer B would cause a syntax error in BASIC."
1650 PRINT"        See part 1, LLIST."
1660 GOTO 1690
1670 PRINT"CORRECT"
1680 Q(2) = 1
1690 PRINT
1700 INPUT"press ENTER";T$
1710 GOSUB 4400
1720 PRINT"You may access the middle record of a sequential file without"
1730 PRINT"reading in the records that are in front of it."
1740 PRINT
1750 PRINT"Is the above statement TRUE or FALSE"
1760 PRINT
1770 PRINT"A  TRUE"
1780 PRINT"B  FALSE"
1790 PRINT
1800 GOSUB 4470
1810 PRINT
1820 IF T$ = "B" THEN GOTO 1880
1830 PRINT"WRONG - the correct answer is B"
1840 PRINT"        The only way to access a record in the middle of the"
1850 PRINT"        file is to read the records in front of it."
1860 PRINT"        See part 1, Sequential Files."
1870 GOTO 1900
1880 PRINT"CORRECT"
1890 Q(3) = 1
1900 PRINT
1910 INPUT"press ENTER";T$
```

```
1920 GOSUB 4400
1930 PRINT"To transfer data from your file to a disk, the computer must "
1940 PRINT"first send the data to a buffer where it is processed, the same"
1950 PRINT"is true for transferring data from the disk back to the"
1960 PRINT"computer."
1970 PRINT
1980 PRINT"Is the above paragraph TRUE or FALSE?"
1990 PRINT
2000 PRINT"A  TRUE"
2010 PRINT"B  FALSE"
2020 PRINT
2030 GOSUB 4470
2040 PRINT
2050 IF T$ = "A" THEN GOTO 2110
2060 PRINT"WRONG - the correct answer is A"
2070 PRINT"       Buffers are used for all data transfer between the"
2080 PRINT"       computer and the disks."
2090 PRINT"       See part 1, Sequential files, and part 2, OPEN & CLOSE"
2100 GOTO 2130
2110 PRINT"CORRECT"
2120 Q(4)=1
2130 PRINT
2140 INPUT"press ENTER";T$
2150 GOSUB 4400
2160 PRINT"You wish to access a previously created disk file.  You will"
2170 PRINT"read in the data and use it to make an important financial"
2180 PRINT"decision.  Which of the following statements will open the file"
2190 PRINT"and read the data into buffer 5.  The filename is B52DAT"
2200 PRINT
2210 PRINT"A  10 OPEN"CHR$(34)"O"CHR$(34)",1,"CHR$(34)"B52DAT"CHR$(34)
2220 PRINT"B  20 OPEN"CHR$(34)"5"CHR$(34)",I,"CHR$(34)"B52DAT"CHR$(34)
2230 PRINT"C  30 OPEN"CHR$(34)"I"CHR$(34)",5,B52DAT"
2240 PRINT"D  40 OPEN"CHR$(34)"I,5,"CHR$(34)","CHR$(34)"B52DAT"CHR$(34)
2250 PRINT"E  None of the above"
2260 PRINT
2270 GOSUB 4470
2280 PRINT
2290 IF T$="E" THEN GOTO 2380
2300 PRINT"WRONG - the correct answer is E"
2310 PRINT"       the correct way is OPEN"CHR$(34)"I"CHR$(34)",5,"CHR$(34)"B52DAT"CHR$(34)
2320 PRINT"       Answer A tries to open the file for output, answer B"
2330 PRINT"       misplaces the 'I' and buffer number, answer C doesn't"
2340 PRINT"       have quotes around the filename, and answer D should"
2350 PRINT"       have quotes only around the 'I' and the filename."
2360 PRINT"       See part 2, OPEN"
2370 GOTO 2400
```

```
2380 PRINT"CORRECT"
2390 Q(5) = 1
2400 PRINT
2410 INPUT"press ENTER";T$
2420 GOSUB 4400
2430 PRINT"What is wrong with the program below (assume the END statement"
2440 PRINT"in line 20 automatically closes the files when it is executed)"
2450 PRINT
2460 PRINT"10 OPEN"CHR$(34)"I"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
2470 PRINT"20 IF EOF(1) THEN END"
2480 PRINT"30 INPUT#1,A$"
2490 PRINT"40 PRINT A$"
2500 PRINT"50 GOTO 10"
2510 PRINT
2520 PRINT
2530 PRINT"A  Nothing"
2540 PRINT"B  A 'File already open' ERROR will be generated"
2550 PRINT"C  The wrong file mode is used"
2560 PRINT"D  Line 20 is illegal"
2570 PRINT
2580 GOSUB 4470
2590 PRINT
2600 IF T$ = "B" THEN GOTO 2650
2610 PRINT"WRONG - the correct answer is B"
2620 PRINT"        Line 50 should be 'GOTO 20' to fix the problem."
2630 PRINT"        See part 2, OPEN and CLOSE"
2640 GOTO 2670
2650 PRINT"CORRECT"
2660 Q(6) = 1
2670 PRINT
2680 INPUT"press ENTER";T$
2690 GOSUB 4400
2700 PRINT"The CLOSE word may be used without a buffer number; however,"
2710 PRINT"when you do that, only the most recent file is CLOSEd."
2720 PRINT
2730 PRINT"Is the above statement TRUE or FALSE?"
2740 PRINT
2750 PRINT"A  TRUE"
2760 PRINT"B  FALSE"
2770 PRINT
2780 GOSUB 4470
2790 PRINT
2800 IF T$ = "B" THEN GOTO 2860
2810 PRINT"WRONG - the correct answer is B"
2820 PRINT"       When the CLOSE word is used without a number, ALL"
2830 PRINT"       the previously opened files are closed."
```

```
2840 PRINT"       See part 2, CLOSE."
2850 GOTO 2880
2860 PRINT"CORRECT"
2870 Q(7) = 1
2880 PRINT
2890 INPUT"press ENTER";T$
2900 GOSUB 4400
2910 PRINT"What is wrong with the following program?"
2920 PRINT
2930 PRINT"10  OPEN"CHR$(34)"O"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
2940 PRINT"20      PRINT#1,10:20:30"
2950 PRINT"30  CLOSE"
2960 PRINT
2970 PRINT
2980 PRINT"A  Nothing"
2990 PRINT"B  Wrong file mode is used"
3000 PRINT"C  The CLOSE statement is invalid"
3010 PRINT"D  The PRINT#1 statement should be INPUT#1"
3020 PRINT
3030 GOSUB 4470
3040 PRINT
3050 IF T$ = "A" THEN GOTO 3090
3060 PRINT"WRONG - the correct answer is A"
3070 PRINT"        See part 2, OPEN and CLOSE"
3080 GOTO 3110
3090 PRINT"CORRECT"
3100 Q(8) = 1
3110 PRINT
3120 INPUT"press ENTER";T$
3130 GOSUB 4400
3140 PRINT"You have opened a file correctly, and you now want to read in "
3150 PRINT"the data from it.  Type in the command you would use."
3160 PRINT"Use line number 10, a space, and the command.  Use buffer #8"
3170 PRINT"and read  the data into variable A$"
3180 PRINT
3190 LINE INPUT"ENTER the command now ? ";T$
3200 PRINT
3210 IF T$ = "10 INPUT#8,A$" THEN GOTO 3250
3220 PRINT"WRONG - the correct answer is : 10 INPUT#8,A$"
3230 PRINT"        See part 2, PRINT#, and INPUT#"
3240 GOTO 3270
3250 PRINT"CORRECT"
3260 Q(9)=1
3270 PRINT
3280 INPUT"press ENTER";T$
3290 GOSUB 4400
```

```
3300 PRINT"What is wrong with the following program?"
3310 PRINT
3320 PRINT"10 IF EOF(1) THEN GOTO 40"
3330 PRINT"20 INPUT#1,A$"
3340 PRINT"30 GOTO 10"
3350 PRINT"40 CLOSE"
3360 PRINT"50 END"
3370 PRINT
3380 PRINT"A  Nothing"
3390 PRINT"B  The file wasn't properly opened"
3400 PRINT"C  Nothing will happen with the data"
3410 PRINT"D  The END statement is not needed"
3420 PRINT
3430 GOSUB 4470
3440 PRINT
3450 IF I$ = "B" THEN GOTO 3500
3460 PRINT"WRONG - the correct answer is B"
3470 PRINT"        The file should be opened before an EOF check is done."
3480 PRINT"        See part 2"
3490 GOTO 3520
3500 PRINT"CORRECT"
3510 Q(10) = 1
3520 PRINT
3530 INPUT"press ENTER";I$
3540 GOSUB 4400
3550 FOR X = 1 TO 10
3560    Y = Y+Q(X)
3570 NEXT X
3580 PRINT"You have finished the test. out of 10 possible correct answers"
3590 PRINT"you scored "Y"."
3600 PRINT
3610 IF Y > 6 THEN  PRINT"YOU HAVE PASSED"
3620 GOSUB 4250
3630 IF Y > 6 THEN GOTO 3710
3640 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
3650 PRINT
3660 PRINT"YOU SHOULD RETAKE LESSON 4!"
3670 PRINT
3680 PRINT"You will be returned to the Menu."
3690 PRINT
3700 GOTO 4490
3710 PRINT
3720 PRINT"Do you want your score recorded on a permanent file?"
3730 PRINT
3740 PRINT"A  YES"
3750 PRINT"B  NO"
```

```
3760 PRINT
3770 INPUT"Which";T$
3780 IF T$ = "B" THEN GOTO 4080
3790 GOSUB 4400
3800 PRINT"To record your score, we must open a file and put your name"
3810 PRINT"in it.  Therefore, surprisingly, we need your name.  If your"
3820 PRINT"name is not unique among the students likely to take this test,"
3830 PRINT"please contact your test monitor for an identifying word that"
3840 PRINT"will make you unique.  Then enter that word below."
3850 PRINT
3860 PRINT"If you have already entered a score previously, be sure to"
3870 PRINT"enter the same name you used before.  (use all capitals!)"
3880 PRINT
3890 INPUT"ENTER your word or name now";T$
3900 OPEN"I",1,"SCORE4"
3910 X = 0
3920 IF EOF(1) THEN GOTO 3980
3930 X = X+1
3940 INPUT#1,N$(X)
3950 INPUT#1,S(X)
3960 IF N$(X) = T$ THEN GOTO 4140
3970 GOTO 3920
3980 CLOSE
3990 X = X+1
4000 N$(X) = T$
4010 S(X) = Y
4020 OPEN"O",1,"SCORE4"
4030 FOR W = 1 TO X
4040   PRINT#1,N$(W)
4050   PRINT#1,S(W)
4060 NEXT W
4070 PRINT
4080 GOSUB 4400
4090 PRINT"You are now qualified to go to LESSON 5."
4100 PRINT
4110 PRINT
4120 INPUT"Do you wish a homework assignment (Y/N)";T$
4122 IF T$ = "N" THEN GOTO 4510
4130 GOTO 4530
4140 S(X) = Y
4150 IF EOF(1) THEN CLOSE:GOTO 4190
4160 X = X+1
4170 INPUT#1, N$(X), S(X)
4180 GOTO 4150
4190 OPEN"O",1,"SCORE4"
4200   FOR W = 1 TO X
```

```
4210        PRINT#1,N$(W)
4220        PRINT#1,S(W)
4230    NEXT W
4240 GOTO 4080
4250 IF T=10 THEN RETURN
4260 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
4270 PRINT
4280 IF Q(1) = 0 THEN PRINT"     LPRINT"
4290 IF Q(2) = 0 THEN PRINT"     LLIST"
4300 IF Q(3) = 0 OR Q(4) = 0 THEN PRINT"     Sequential Files"
4310 IF Q(5) = 0 OR Q(6) = 0 OR Q(7) = 0 THEN PRINT"     OPEN and CLOSE"
4320 IF Q(8) = 0 OR Q(9) = 0 OR Q(10) = 0 THEN PRINT"     PRINT#, INPUT#, and EOF"
4330 PRINT
4340 INPUT"press ENTER";I$
4350 GOSUB 4410
4360 RETURN
4370 REM **
4380 REM ** this subroutine clears the screen"
4390 REM **
4400 FOR X = 1 TO 24
4410     PRINT
4420 NEXT X
4430 RETURN
4440 REM **
4450 REM ** this subroutine is for the response section
4460 REM
4470 INPUT"ENTER the letter opposite the correct answer";I$
4480 RETURN
4490 PRINT
4500 INPUT"press ENTER to return to MENU";I$
4510 RUN"MENU"
4520 CLOSE
4530 OPEN"I",1,"SCORE4"
4540  IF EOF(1) THEN STOP
4550 INPUT#1,A$,N
4560 PRINTA$,N
4570 GOTO 4540
4580 RUN "HW4"
```

```
***** Listing of Program 'LESSON5' *****                    07/10/83 - 03:20:11

 10 REM ** THIS PROGRAM STARTED ON 7 JUN 1983
 20 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
 30 REM ** TITLE: LESSON 5
 40 REM **
 50 REM **
100 REM **
190 REM **
200 GOSUB 21000
210 PRINT"LESSON:  BASIC 5            VERSION: 1 AUGUST 83
215 PRINT
220 PRINT"TIME REQUIRED TO COMPLETE LESSON:  About one hour"
250 PRINT
280 PRINT
290 PRINT"AUTHOR:  Capt Danny J. Creagan"
700 PRINT"         Air Force Institute of Technology"
710 PRINT
720 PRINT"OBJECTIVE:  To teach the student how to use SUBROUTINES "
730 PRINT"            and LIBRARY functions."
735 PRINT
740 PRINT
741 PRINT
742 PRINT
744 PRINT
750 INPUT"Press the ENTER key to continue";T$
755 GOSUB 21000
760 GOSUB 30000
870 PRINT"A  I'm taking this part in its entirety."
880 PRINT"B  I wish to review selected areas."
902 PRINT"C  I want to go to the second part."
904 PRINT"D  I want to return to the Menu."
910 PRINT
915 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
920 IF T$ = "D" GOTO 49000
922 IF T$ = "C" GOTO 49000
930 IF T$ = "B" GOTO 1000
940 IF T$ = "A" GOTO 990
950 GOSUB 2000
955 GOSUB 3000
960 GOSUB 4000
970 GOSUB 5000
975 GOSUB 8000
980 GOSUB 21000
985 PRINT"You are now done with this part of the lesson.  If you ENTER a "
986 PRINT"'C' you will go on to the second part.  ENTER an 'R' to"
987 INPUT"start over.           Which do you want C or R";T$
988 IF T$ = "R" THEN RUN
```

```
980 IF T$ <> "C" THEN GOTO 980
990 GOTO 49000
1000 GOSUB 21000
1002 GOSUB 31000
1010 PRINT
1015 PRINT"Please type in the number beside the area you wish"
1010 PRINT"to review (1 through 5) and then press ENTER - press 0 and"
1015 PRINT"press ENTER to return to the Menu."
1025 PRINT
1030 INPUT"What is your choice";N
1040 IF N = 0 GOTO 45000
1050 ON N GOSUB 2000,3000,4000,5000,6000
1060 GOTO 1000
2000 GOSUB 21000
2010 PRINT"                              INTRODUCTION"
2020 PRINT
2030 PRINT"In the last episode of our computerized book, we assigned you a"
2040 PRINT"homework problem that dealt with disk I/O (disk input/output)."
2050 PRINT"It included most of the concepts that we have been studying."
2060 PRINT"Now we are on the downhill part of the course. That's right,
2070 PRINT"you are almost done with the hard parts of BASIC."
2095 PRINT
2100 PRINT"The last obstacle is learning about SUBROUTINES, which you will"
2110 PRINT"learn in the first half of this lesson.  After that, we will"
2120 PRINT"review the numerous library functions (but not have to memorize"
2130 PRINT"them, we'll just have to know how they work), and then go on"
2140 PRINT"to lesson 6."
2150 PRINT
2160 INPUT"press ENTER";T$
2170 GOSUB 21000
2180 PRINT"                              INTRODUCTION"
2190 PRINT
2200 PRINT"Somewhere during each lesson, we emphasize the value of doing"
2210 PRINT"extra study, besides what this course teaches you.  We all learn"
2220 PRINT"more when we DO something that we have read about.  This is"
2230 PRINT"certainly true with learning a programming language."
2240 PRINT
2250 PRINT"For now, though, get out your favorite BASIC manual, and curl"
2260 PRINT"up beside your computer for another lesson in BASIC!"
2270 PRINT
2280 INPUT"press ENTER";T$
2290 GOSUB 41000
2300 IF T$ = "B" THEN GOTO 2000
2310 RETURN
3000 GOSUB 21000
3010 PRINT"                              SUBROUTINES"
```

```
 3020 PRINT
 3030 PRINT"We use the word 'ROUTINE' to describe the statements in the"
 3040 PRINT"body of a program.  'SUBROUTINE' is used to describe"
 3050 PRINT"a miniprogram that was built, attached to the main program,"
 3060 PRINT"and used to perform a mini-task that, for some reason,"
 3070 PRINT"needs special attention."
 3080 PRINT
 3090 PRINT"Usually, we use subroutines to do tasks which we perform"
 3100 PRINT"often in our program such as printing out a menu after"
 3110 PRINT"each module of a CAI program is finished by a student."
 3120 PRINT"That way, we only have to write the menu program once and call"
 3125 PRINT"on it when we need it."
 3130 PRINT
 3140 INPUT"press ENTER";T$
 3141 GOSUB 21000
 3142 PRINT"Why would we want to use a subroutine?"
 3143 PRINT
 3144 PRINT"A  To do those parts of the program that are used often"
 3145 PRINT"B  To go UNDER the normal routines"
 3146 PRINT
 3147 INPUT"ENTER the letter opposite the correct answer";T$
 3148 PRINT
 3149 IF T$ = "A" THEN GOTO 3155
 3150 PRINT"WRONG - this was supposed to be an easy question to answer."
 3151 PRINT"Obviously, we screwed up somehow.  We will send you back to the"
 3152 PRINT"beginning of this part. Try reading between the lines a little."
 3153 PRINT:INPUT"press ENTER";T$
 3154 GOTO 3000
 3155 PRINT"CORRECTO - GOOD JOB!"
 3156 PRINT
 3157 INPUT"press ENTER";T$
 3158 GOSUB 21000
 3160 PRINT"                              SUBROUTINES"
 3170 PRINT
 3180 PRINT"Subroutines are mini-programs that we put in our main program"
 3190 PRINT"and, when we need them, we can go to them, perform the"
 3200 PRINT"task they are built to do, and return to the main program."
 3210 PRINT
 3220 PRINT"Subroutines differ from other forms of program control in"
 3230 PRINT"that they ALWAYS RETURN CONTROL TO THE STATEMENT THAT"
 3240 PRINT"FOLLOWS THE STATEMENT THAT CALLED THEM. That means you"
 3250 PRINT"can call a subroutine anywhere within a program, and the"
 3260 PRINT"computer will remember where the call came from, and when"
 3270 PRINT"the subroutine has done its job, control will go back to"
 3280 PRINT"the statement that followed the calling statement."
 3290 PRINT
```

```
3340 PRINT
3350 INPUT"press ENTER";T$
3360 GOSUB 21000
3365 PRINT"                              SUBROUTINES"
3367 PRINT
3370 PRINT"If you have built a program that has to continually print"
3380 PRINT "out a prompt, asking the users if they want to review"
3390 PRINT"previous sections of the program that have run, you would"
3400 PRINT"likely use a subroutine to ask the question, return to"
3410 PRINT"the main program with the answer stored in a variable,"
3420 PRINT"and branch to the right part of the program, based on the"
3430 PRINT"answer."
3435 PRINT
3440 PRINT
3440 PRINT"The following is an example such a program"
3450 PRINT
3460 INPUT"press ENTER";T$
3465 GOSUB 21000
3470 GOSUB 41000
3480 PRINT
3490 PRINT"We will explore the main points of this routine in the next "
3500 INPUT"section.  Press ENTER to go on";T$
3510 PRINT
3530 GOSUB 21000
3540 PRINT"Where does control transfer when a subroutine returns to the"
3550 PRINT"main program?"
3560 PRINT
3570 PRINT"A  The beginning of the program"
3580 PRINT"B  The calling statement"
3590 PRINT"C  The statement after the calling statement"
3600 PRINT"D  None of the above"
3610 PRINT
3620 INPUT"Enter the letter opposite the correct answer";T$
3630 PRINT
3640 IF T$ = "C" THEN GOTO 3680
3650 PRINT"WRONG - control is transferred to the statement after the "
3660 PRINT"        calling statement.  The correct answer is C"
3670 GOTO 3690
3680 PRINT"CORRECT - Super job . . . Now we can go on"
3690 PRINT
3700 INPUT"press ENTER";T$
3710 GOSUB 40000
3720 IF T$ = "B" GOTO 3000
3730 RETURN
4000 GOSUB 21000
4005 PRINT"                              GOSUB & RETURN"
```

```
4010 PRINT
4020 PRINT"The set of statements that you use to implement a subroutine"
4030 PRINT"is made of the GOSUB and RETURN words.  The GOSUB word is used"
4040 PRINT"almost exactly like the GOTO statement.  You put the line"
4050 PRINT"number of the start of the subroutine on the right of the GOSUB"
4060 PRINT"word.  When the computer gets to it, it transfers control to"
4070 PRINT"the subroutine.  When the subroutine is done, it returns to the"
4080 PRINT"main program by using the RETURN statement.  You cannot GOTO"
4090 PRINT"the main program from a subroutine without risking disaster."
4100 PRINT"You should always use the RETURN statement."
4110 PRINT
4120 INPUT"press ENTER";T$
4125 GOSUB 21000
4130 GOSUB 41000
4140 PRINT
4150 PRINT"Line 20 and line 90 call the subroutine, and line 10010 RETURNs"
4160 INPUT"to the appropriate statement.    press ENTER";T$
4170 GOSUB 21000
4180 GOSUB 41000
4190 PRINT
4200 PRINT"If line 20 calls the subroutine, what line gets control after"
4210 INPUT"the RETURN statement .... ENTER the correct line number";T$
4220 PRINT
4240 IF T$ = "50" THEN GOTO 4260
4250 PRINT"WRONG - the correct answer is line 50"
4255 GOTO 4262
4260 PRINT"CORRECT - You are getting GOOOOOOOgood!"
4262 PRINT
4265 INPUT"press ENTER";T$
4270 GOSUB 21000
4280 PRINT"                              GOSUB & RETURN"
4290 PRINT
4300 PRINT"You can have more than one GOSUB in a program, and you can"
4310 PRINT"have more than one RETURN in a subroutine.  If you have more"
4320 PRINT"than one RETURN, then the computer will return when it reaches"
4330 PRINT"the first RETURN statement it comes to.  Generally speaking,"
4340 PRINT"you should try to limit the number of exits from a subroutine"
4350 PRINT"because it can get very confusing if you have RETURNs stuck"
4360 PRINT"all over the place.  It is usually possible to have only one"
4370 PRINT"exit to any program or subroutine."
4380 PRINT
4390 INPUT"press ENTER";T$
4400 GOSUB 21000
4410 PRINT"Is the following program valid?"
4420 PRINT
4440 PRINT"20 INPUT"CHR$(34)"Enter a number between 0 and 20"CHR$(34)";N
```

```
4450 PRINT"30 GOSUB 100"
4460 PRINT"40 END"
4470 PRINT"100 IF N < 10 THEN PRINT "CHR$(34)"Number is less than 10"CHR$(34)" ELSE GOTO 120"
4480 PRINT"110 RETURN"
4490 PRINT"120 IF N > 10 THEN PRINT "CHR$(34)"Number is more than 10"CHR$(34)" ELSE GOTO 140"
4500 PRINT"130 RETURN"
4510 PRINT"140 PRINT "CHR$(34)"Your number is 10"CHR$(34)
4520 PRINT"150 RETURN"
4530 PRINT
4540 PRINT"A  Yes, but confused by too many RETURNs"
4545 PRINT"B  No, the program will never get passed line 110"
4550 PRINT
4560 INPUT"ENTER the letter opposite the correct answer";T$
4570 PRINT
4580 IF T$ = "A" THEN GOTO 4610
4590 PRINT"WRONG - the only problem is a confusing number of RETURNs"
4600 GOTO 4620
4610 PRINT"CORRECT"
4620 PRINT
4630 INPUT"press ENTER";T$
4640 GOSUB 21000
4650 PRINT"ENTER the command to go to a subroutine that starts on line"
4660 PRINT"1000.  Use line 100 and put one space between all terms."
4670 PRINT
4680 INPUT"Enter the command now";T$
4690 PRINT
4700 IF T$ = "100 GOSUB 1000" THEN GOTO 4730
4710 PRINT"WRONG - the correct answer is : 100 GOSUB 1000"
4720 GOTO 4750
4730 PRINT"CORRECT"
4750 PRINT
4760 INPUT"press ENTER";T$
4770 GOSUB 40000
4780 IF T$ = "B" THEN GOTO 4000
4790 RETURN
5000 GOSUB 21000
5010 PRINT"                              NESTED SUBROUTINES"
5020 PRINT
5030 PRINT"What do you think the output of the following is?"
5040 PRINT
5050 GOSUB 42000
5060 PRINT
5070 INPUT"press ENTER for the answer";T$
5080 GOSUB 21000
5090 PRINT"The answer is :"
5100 GOSUB 42000
```

```
5110 PRINT
5120 PRINT"Main Program"
5130 PRINT"Subroutine One"
5140 PRINT"Subroutine Two"
5150 PRINT"Subroutine Two"
5160 INPUT"                              press ENTER";T$
5170 GOSUB 21000
5180 GOSUB 42000
5190 PRINT
5200 PRINT"This is an example of a 'NESTED SUBROUTINE'.  The subroutine"
5210 PRINT"in line 100 calls the subroutine in line 200.  It is perfectly"
5220 PRINT"legal, and sometimes very valuable. "
5230 PRINT
5240 INPUT"press ENTER";T$
5250 GOSUB 21000
5260 PRINT"What is the order of execution of the following program?"
5270 PRINT
5280 PRINT"10 GOSUB 100"
5290 PRINT"20 GOSUB 200"
5300 PRINT"30 END"
5310 PRINT"100 RETURN"
5320 PRINT"200 GOSUB 100"
5330 PRINT"210 RETURN"
5340 PRINT
5350 PRINT"Type out the line numbers as they would be executed, putting a"
5360 PRINT"space between each number. (ie 10 20 30 40)"
5370 INPUT"Enter your answer";T$
5380 PRINT
5390 IF T$ = "10 100 20 200 100 210 30" THEN GOTO 5420
5400 PRINT "WRONG - the correct answer is :10 100 20 200 100 210 30"
5405 PRINT"       That was a rough one, you may want to review this part"
5410 GOTO 5430
5420 PRINT"CORRECT - gets complicated doesn't it?"
5430 PRINT
5440 INPUT"press ENTER";T$
5450 GOSUB 40000
5460 IF T$ = "B" THEN GOTO 5000
5470 RETURN
8000 GOSUB 21000
8010 PRINT"                                  ON GOSUB"
8020 PRINT
8030 PRINT"Remember the ON GOTO statement from our previous lessons?"
8040 PRINT
8050 PRINT"There's a command very similar to ON GOTO that can be used with"
8060 PRINT"subroutines.  You have probably already guessed that the "
8070 PRINT"command is called ON GOSUB."
```

```
8090 PRINT"Here is an example:"
8100 PRINT
8110 PRINT"10 INPUT"CHR$(34)"ENTER A NUMBER BETWEEN 1 AND 4"CHR$(34)";N"
8120 PRINT"20 ON N GOSUB 300,400,500,600"
8130 PRINT
8140 PRINT"If N = 1 then subroutine 300 would be executed. if N = 2, sub-"
8150 PRINT"routine 400 would execute, and so on"
8160 PRINT
8170 INPUT"PRESS ENTER";T$
8180 GOSUB 21000
8190 PRINT"                              ON GOSUB"
8200 PRINT
8210 PRINT"10 INPUT"CHR$(34)"ENTER A NUMBER BETWEEN 1 AND 4"CHR$(34)";N"
8220 PRINT"20 ON N GOSUB 300,400,500,600"
8230 PRINT
8240 PRINT"If the value of N exceeds the number of options that are"
8250 PRINT"available, then ON GOSUB will default to the first available"
8260 PRINT"line number (in this case it would be 300)"
8270 PRINT
8280 PRINT
8290 INPUT"PRESS ENTER FOR AN EXAMPLE";T$
8300 GOSUB 21000
8310 PRINT"10 PRINT"CHR$(34)"TYPE 1 FOR SQUARE TABLE"CHR$(34)
8320 PRINT"20 PRINT"CHR$(34)"TYPE 2 FOR THE CUBE TABLE"CHR$(34)
8330 PRINT"30 INPUT A"
8340 PRINT"40 ON A GOSUB 1000,2000"
8350 PRINT"50 GOTO 10"
8360 PRINT"1000 FOR X = 1 TO 50"
8370 PRINT"1010   PRINT X,X*X"
8380 PRINT"1020 NEXT X"
8390 PRINT"1030 RETURN"
8400 PRINT"2000 FOR X = 1 TO 50"
8410 PRINT"2010   PRINT X, X*X*X"
8420 PRINT"2020 NEXT X"
8430 PRINT"2030 RETURN"
8440 PRINT
8450 PRINT"If you ENTER a 1, then subroutine 1000 is used.  Which subrou-"
8460 INPUT"tine is used if you ENTER a 4 (1000,2000 or NONE)";T$
8470 PRINT
8480 IF T$ = "NONE" THEN GOTO 8510
8490 PRINT"WRONG - THE CORRECT ANSWER IS NONE"
8500 GOTO 8520
8510 PRINT"CORRECT"
8520 PRINT
8530 INPUT"PRESS ENTER";T$
8540 GOSUB 43000
```

```
 3550 IF T$ = "B" THEN GOTO 9000
 3560 RETURN
20980 REM **
20990 REM ** This subroutine clears the screen on any terminal
20995 REM **
21000 FOR X = 1 TO 24
21010    PRINT
21020 NEXT X
21030 RETURN
22000 REM **
22020 REM **  this subr is the menu
22030 REM **
30000 PRINT"                    LESSON 5"
30010 PRINT
30015 PRINT"This is the first part of a two part lesson"
30020 PRINT"It is divided into the following sections."
30025 PRINT
30030 PRINT"1)  Introduction          4)  Nested Subroutines"
30035 PRINT"2)  Subroutines           5)  ON GOSUB"
30040 PRINT"3)  GOSUB & RETURN"
30060 PRINT
30090 PRINT
30100 RETURN
30200 REM **
30300 REM ** rem ** this subr gives the student a chance to review the lesson
30400 REM **
40000 GOSUB 21000
40005 PRINT"Which do you wish to do?"
40010 PRINT
40020 PRINT"A  Continue on"
40030 PRINT"B  Review this section again"
40040 PRINT
40050 INPUT"press the letter opposite the correct answer and press ENTER";T$
40060 IF T$ = "A" OR T$ = "B" THEN RETURN
40070 GOTO 40000
40970 REM **
40980 REM ** this subroutine is for the gosub example
40990 REM **
41000 PRINT"10 print"CHR$(34)"A CAI program is a computer assisted instruction"CHR$(34)
41010 PRINT"20 print"CHR$(34)"program that teaches students."CHR$(34)
41030 PRINT"30 GOSUB 10000     :'<---- LOOK use sub. after every major part"
41040 PRINT"50 IF T$ = "CHR$(34)"B"CHR$(34)" THEN GOTO 10 :' T$ returns from sub with choice"
41050 PRINT"60 .....ETC"
41057 PRINT"80 GOSUB 10000     :'<---- LOOK use sub. after every major part"
41058 PRINT"90 END   :'You must ensure program doesn't goto sub by accident"
41059 PRINT"100 REM"
```

```
41060 PRINT"110 REM the subroutine follows"
41063 PRINT"120 REM"
41065 PRINT"10000 INPUT"CHR$(34)"Do you want to continue or review (A=Con B=Rev)"CHR$(34)";T$"
41070 PRINT"10010 RETURN"
41080 RETURN
42000 PRINT"10 PRINT"CHR$(34)"Main Program"CHR$(34)
42010 PRINT"20 GOSUB 100"
42020 PRINT"30 GOSUB 200"
42030 PRINT"40 END"
42040 PRINT"100 PRINT"CHR$(34)"Subroutine One"CHR$(34)
42045 PRINT"110 GOSUB 200"
42050 PRINT"120 RETURN"
42060 PRINT"200 PRINT"CHR$(34)"Subroutine Two"CHR$(34)
42070 PRINT"210 RETURN"
42080 RETURN
48000 RUN "MENU"
49000 RUN"lesson5a"
50000 END
```

```
 755 GOSUB 21000
 760 GOSUB 30000
 890 PRINT"A  I'm taking this part in its entirety."
 900 PRINT"B  I wish to review selected areas (or take the test)."
 902 PRINT"C  I want to go to the first part."
 904 PRINT"D  I want to return to the Menu."
 910 PRINT
 915 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
 920 IF T$ = "D" GOTO 48000
 922 IF T$ = "C" GOTO 49000
 930 IF T$ = "B" GOTO 1000
 940 IF T$ = "A" GOTO 990
 950 GOSUB 2000
 955 GOSUB 3000
 960 GOSUB 4000
 965 GOSUB 5000
 977 GOSUB 7000
 978 GOSUB 8020
 980 GOSUB 21000
 990 GOTO 49000
1000 GOSUB 21000
1002 GOSUB 30000
1005 PRINT
1005 PRINT"Please type in the number beside the area you wish"
1010 PRINT"to review (1 through 8) and then press ENTER - press 0 and"
1015 PRINT"press ENTER to return to the Menu."
1025 PRINT
1030 INPUT"What is your choice";N
1040 IF N = 0 GOTO 48000
1050 ON N GOSUB 2000,3000,4000,5000,7000,47000
1060 GOTO 1000
2000 GOSUB 21000
2010 PRINT"                              INTRODUCTION"
2020 PRINT
2030 PRINT"In this lesson we will tackle the different catagories and"
2040 PRINT"kinds of FUNCTIONs that are available to us in Microsoft BASIC."
2050 PRINT"We will use principles already covered in previous"
2070 PRINT"sessions.  If you have trouble with some of the concepts, then"
2080 PRINT"you will have to review the appropriate lesson to catch up."
2090 PRINT
2100 PRINT"However, we will not be going into great depth with our"
2110 PRINT"explanations.  Many FUNCTIONs have very specific uses and it"
2120 PRINT"would not be useful for you to memorize them.  The idea you"
2130 PRINT"should get from this part is that there are many functions"
2140 PRINT"available, and that when you need them, you should get out your"
2150 PRINT"manual and look up the specific implementation of each one."
```

```
 2160 PRINT
 2170 INPUT"press ENTER";T$
 2180 RETURN
 3000 GOSUB 21000
 3010 PRINT"                        Functions Overview"
 3020 PRINT
 3030 PRINT"Remember in the second part of Lesson 1 when we described"
 3040 PRINT"FUNCTIONs for the first time?  We said that we would come back"
 3050 PRINT"to them in another lesson.  Well, this is it!"
 3060 PRINT
 3070 PRINT"From that lesson we should remember that functions are pre-"
 3080 PRINT"written instructions that perform commonly used operations."
 3090 PRINT"You can look at functions like they were mini-subroutines,"
 3100 PRINT"only you don't use GOSUB or RETURN statements to call them!"
 3110 PRINT"Instead, you just use the keyword associated with the FUNCTION"
 3120 PRINT"and the computer performs the appropriate operation auto-"
 3130 PRINT"matically.  In the next sections we will study two types of"
 3140 PRINT"functions, Library, and User Functions."
 3150 PRINT
 3160 INPUT"press ENTER";T$
 3170 GOSUB 21000
 3180 PRINT"                              Functions Overview"
 3190 PRINT
 3200 PRINT"Library functions contain useful operations that have been"
 3210 PRINT"written and stored in the computer, and are there whenever you"
 3220 PRINT"need them.  User functions are functions that you make up"
 3230 PRINT"by inserting the instruction to make them in your program."
 3240 PRINT"Then, when your program needs the special USER FUNCTION, it"
 3250 PRINT"can call on it with a special word."
 3260 PRINT
 3270 PRINT"There are many functions, and, depending on the specific"
 3280 PRINT"implementation of Microsoft BASIC, you probably have at least"
 3290 PRINT"12 Library functions stored in your computer.  On the next"
 3300 PRINT"screen is a list of the typical set of Library functions."
 3310 PRINT
 3320 INPUT"press ENTER";T$
 3330 GOSUB 21000
 3340 PRINT"                        Typical Library Functions"
 3350 PRINT
 3360 GOSUB 41000
 3370 PRINT
 3380 PRINT"Specific examples of these functions will be given later, or"
 3390 PRINT"you may look them up in your BASIC manual."
 3400 INPUT"press ENTER";T$
 3410 GOSUB 21000
 3420 PRINT"Is the following statement TRUE or FALSE?"
```

```
3430 PRINT
3440 PRINT"Library functions are predefined instructions that are stored"
3450 PRINT"inside the computer, and may be used at any time in your"
3460 PRINT"program."
3470 PRINT
3480 PRINT"A  TRUE"
3490 PRINT"B  FALSE"
3500 PRINT
3510 INPUT"ENTER the letter opposite the correct answer";T$
3520 PRINT
3530 IF T$ = "A" THEN GOTO 3560
3540 PRINT"WRONG - the sentence is good, correct answer is A"
3550 GOTO 3570
3560 PRINT"CORRECT"
3570 PRINT
3580 INPUT"press ENTER";T$
3590 GOSUB 21000
3600 PRINT"Is the following sentence TRUE or FALSE?"
3610 PRINT
3620 PRINT"User Functions are made by the user and are not normally stored"
3630 PRINT"in the computer."
3640 PRINT
3650 PRINT"A  TRUE"
3660 PRINT"B  FALSE"
3670 PRINT
3680 INPUT"ENTER the letter opposite the correct answer";T$
3690 PRINT
3700 IF T$ = "A" THEN GOTO 3730
3710 PRINT"WRONG - the statement is true, the correct answer is A"
3720 GOTO 3740
3730 PRINT"CORRECT"
3740 PRINT
3750 INPUT"press ENTER";T$
3760 GOSUB 40000
3770 IF T$ = "B" THEN GOTO 3000
3780 RETURN
4000 GOSUB 21000
4010 PRINT"                    Library Functions"
4030 GOSUB 41000
4040 PRINT
4050 PRINT"Here are twelve of the most used library functions.  As an "
4060 PRINT"example of how much time they can save, think of the number of"
4070 PRINT"statements you would have to write to calculate the logarithm"
4080 PRINT"of a number."
4120 INPUT"press ENTER";T$
4130 GOSUB 21000
```

```
4140 PRINT"                          Library Functions"
4150 PRINT
4160 PRINT"All you have to do to calculate the log of a number, say the"
4170 PRINT"number is 10, is to type in the statement - PRINT LOG(10)'"
4190 PRINT"The example of the logaritha of 10 is exactly how we implement"
4200 PRINT"the library functions.  Here is another example:"
4210 PRINT
4220 GOSUB 42000
4250 PRINT
4260 PRINT"In this example, any positive number will have its square root"
4270 PRINT"printed out.  Any negative number will cause an error.  You"
4280 PRINT"can't take the square root of a negative number."
4290 PRINT
4300 INPUT"press ENTER";T$
4310 GOSUB 21000
4320 PRINT"                          Library Functions"
4330 PRINT
4340 GOSUB 42000
4350 PRINT
4360 PRINT"Notice that the function allows you to use a variable as an"
4370 PRINT"argument for the number you want the square root of.  In "
4380 PRINT"addition to this, you can use a FUNCTION as an argument for a"
4390 PRINT"FUNCTION!"
4400 PRINT
4410 INPUT"press ENTER for an example";T$
4420 GOSUB 21000
4430 PRINT"                          Library Functions"
4440 PRINT
4450 GOSUB 43000
4460 PRINT
4470 PRINT"In this example, no matter what the sign of the number you "
4480 PRINT"enter, you will get a valid square root, because ABS(N) will"
4490 PRINT"give the absolute value of N (N as a positive number), and then"
4500 PRINT"SQR(ABS(N)) will give the square root."
4510 PRINT
4520 PRINT"You can use just about any valid numeric expression for the"
4530 PRINT"(exp) part of any Library Function."
4531 PRINT
4532 INPUT press ENTER";T$
4537 GOSUB 21000
4541 PRINT"Is the following statement valid?"
4542 PRINT
4543 PRINT"10 PRINT LOG(0)"
4544 PRINT
4545 PRINT"HINT: If you are not sure, try it on a calculator"
4546 PRINT
```

```
4547 PRINT"A  Yes, it is valid"
4548 PRINT"B  No. 0 is not a valid number for this function"
4549 PRINT
4550 INPUT"ENTER the letter opposite the correct answer";T$
4551 PRINT
4552 IF T$ = "B" THEN GOTO 4556
4553 PRINT"WRONG - you can't take the log of 0"
4554 GOTO 4557
4556 PRINT"CORRECT"
4557 PRINT
4558 INPUT"press ENTER";T$
4560 GOSUB 21000
4570 PRINT"                          Library Functions"
4580 PRINT
4590 GOSUB 43000
4595 PRINT
4600 PRINT"The expression ABS(N) is evaluated first, then the outside"
4610 PRINT"expression - SQR(---) - is evaluated next.  Remember from the"
4620 PRINT"first lesson when we said that the parenthesis is the highest"
4630 PRINT"priority arithmetic expression?  That means that any expression"
4640 PRINT"that is within parenthesis will be evaluated first.  If more"
4650 PRINT"than one set of parenthesis is used, then the expression with-"
4660 PRINT"in the inner-most set of parenthesis is evaluated first."
4670 PRINT
4680 INPUT"press ENTER";T$
4690 GOSUB 21000
4700 PRINT"Is the following sentence TRUE or FALSE?"
4710 PRINT
4720 PRINT"You can write your own library functions in special cases."
4730 PRINT
4740 PRINT"A  TRUE"
4750 PRINT"B  FALSE"
4760 PRINT
4770 INPUT"ENTER the letter opposite the correct answer";T$
4780 PRINT
4790 IF T$ = "B" THEN GOTO 4830
4800 PRINT"WRONG - library functions are permanently stored in the "
4810 PRINT"        computer and cannot be created, the correct answer is B"
4820 GOTO 4840
4830 PRINT"CORRECT"
4840 PRINT
4850 INPUT"press ENTER";T$
4860 GOSUB 21000
4870 PRINT"In the next few screens, we will ask you questions concerning"
4880 PRINT"library functions.  You should get out your BASIC manual and"
4890 PRINT"look up the answer to the questions before you attempt to enter"
```

```
4900 PRINT"your response."
4910 PRINT
4920 PRINT"You should give your answer in the format, FUNCTION(number)."
4930 PRINT"You will always be given the number for the function, and you"
4940 PRINT"do not have to include a line number.  For example, if we ask"
4950 PRINT"for the function that gives the square root of 10, you would"
4960 PRINT"type in: SQR(10)."
4970 PRINT
4980 INPUT"press ENTER";T$
4990 GOSUB 21000
5000 INPUT"What is the function that gives the natural log of 8";T$
5005 PRINT
5010 IF T$ = "LOG(8)" THEN PRINT "CORRECT" ELSE PRINT "WRONG - ANSWER is LOG(8)"
5020 PRINT
5030 INPUT"press ENTER";T$
5040 GOSUB 21000
5050 INPUT"Name the function that gives the absolute value of -3";T$
5060 PRINT
5070 IF T$ = "ABS(-3)" THEN PRINT"CORRECT" ELSE PRINT"WRONG - the answer is ABS(-3)"
5080 PRINT
5090 INPUT"press ENTER";T$
5100 GOSUB 21000
5110 INPUT"What is the function that gives the arctangent of A*3";T$
5120 PRINT
5130 IF T$ = "ATN(A*3)" THEN PRINT"CORRECT - GREAT JOB" ELSE PRINT"WRONG - the correct answer is ATN(A*3)"
5140 PRINT
5150 INPUT"press ENTER";T$
5160 GOSUB 21000
5170 INPUT"What is the function that gives the sine of .5";T$
5180 PRINT
5190 IF T$ = "SIN(.5)" THEN PRINT"CORRECT" ELSE PRINT"WRONG - the correct answer is SIN(.5)"
5200 PRINT
5210 INPUT"press ENTER";T$
5220 GOSUB 21000
5230 INPUT"Name the function to give a random number between 0 & 1";T$
5240 PRINT
5250 IF T$ = "RND(0)" THEN PRINT"CORRECT - good one" ELSE PRINT"WRONG - the correct answer is RND(0)"
5260 PRINT
5270 INPUT"press ENTER";T$
5280 GOSUB 40000
5290 IF T$ = "B" THEN GOTO 4000
5300 RETURN
6000 GOSUB 21000
6010 PRINT"                    User Functions"
6020 PRINT
6030 PRINT"User defined functions let you make up your own functions"
```

```
6040 PRINT"when you can't find a library function that will do the job."
6050 PRINT
6060 PRINT"The statement you use to DEFine a user function, so the"
6070 PRINT"computer will know what you are doing, is called the DEF state-"
6080 PRINT"ment.  The user function only applies to the program in which"
6090 PRINT"it was defined.  When the program is abandoned, the function is"
6100 PRINT"no longer valid.  An example of a user defined function is "
6110 PRINT"coming up .... but first, something a little different."
6120 PRINT
6130 PRINT
6131 INPUT"press ENTER";T$
6132 GOSUB 21000
6141 PRINT"Can user functions be carried over from one program to another?"
6142 PRINT
6143 PRINT"A  Yes"
6144 PRINT"B  No, they always have to be redefined"
6145 PRINT"C  Both A & B above."
6146 PRINT
6147 INPUT"Enter the letter opposite the correct answer";T$
6148 PRINT
6149 IF T$ = "B" THEN GOTO 6155
6150 IF T$ = "C" THEN GOSUB 21000:PRINT"Are you related to my wife?":PRINT"Please make another choice.....
     In fact, try answer B":PRINT:INPUT"press ENTER";T$:GOTO 6141
6153 PRINT"WRONG - the correct answer is B"
6154 GOTO 6156
6155 PRINT"CORRECT"
6156 PRINT
6157 INPUT"press ENTER";T$
6158 GOSUB 21000
6160 PRINT"                    User Functions"
6170 PRINT
6180 GOSUB 44000
6190 PRINT
6200 PRINT"There are two user functions in this program.  They are defined"
6210 PRINT"in lines 10 and 20, and called on in lines 30, and 50.  We will"
6220 PRINT"cover this program in greater detail in the next section."
6230 PRINT
6240 INPUT"press ENTER";T$
6250 GOSUB 40000
6260 IF T$ = "B" THEN GOTO 6000
6270 RETURN
7000 GOSUB 21000
7010 PRINT"                         DEF Statement"
7030 PRINT"The DEF statement is used to define a user function and it has"
7040 PRINT"the following format:"
7050 PRINT
```

```
7060 PRINT"   (Line #)DEF FN(func. name)((vars)) = (exp)"
7070 PRINT
7080 PRINT"You must never use the DEF statement in the IMMEDIATE mode."
7090 PRINT"You always have to have a line number.  In CPM and Cromemco"
7100 PRINT"systems, you must separate the terms DEF and FN by one space."
7110 PRINT"In TRS-80s you don't have to.  The (func. name) is any valid"
7120 PRINT"variable name. ((vars)) is an optional parameter that may be"
7130 PRINT"passed to the function.  If it is used in the DEF statement, it"
7140 PRINT"MUST be used when it is called on. (more on that later). The"
7150 PRINT"term (exp) is the calculation that you wish the function to do."
7160 PRINT
7170 INPUT"press ENTER";T$
7180 GOSUB 21000
7210 GOSUB 44000
7220 PRINT
7230 PRINT"Get out your BASIC manual and look up DEF.  It will show you"
7240 PRINT"examples similar to this.  Line 10 defines a function named"
7250 PRINT"R2 that will be set equal to the expression on the right side"
7260 PRINT"of the statement.  - it will return a random number between 1"
7270 PRINT"and 10.  Note that this user function uses a library function"
7280 PRINT"as part of its definition.  This is legal."
7290 PRINT
7300 INPUT"press ENTER";T$
7301 GOSUB 21000
7302 PRINT"Can you use a library function as part of the definition of a "
7303 PRINT"USER function?"
7304 PRINT
7305 PRINT"A  YES"
7306 PRINT"B  NO"
7307 PRINT
7308 INPUT"ENTER the letter opposite the correct answer";T$
7309 PRINT
7310 IF T$ = "A" THEN GOTO 7315
7311 PRINT"WRONG - you CAN use a library function inside a USER function"
7312 GOTO 7317
7315 PRINT"CORRECT"
7317 PRINT
7318 INPUT"press ENTER";T$
7319 GOSUB 21000
7320 GOSUB 44000
7330 PRINT
7340 PRINT"Line 30 calls on the function defined in line 10.  When it"
7350 PRINT"does, X is set equal to a random number between 1 and 10 and"
7360 PRINT"then it is printed out in line 40.  Line 50 calls on the "
7370 PRINT"function in line 20, but it sends two values to the DEF state-"
7380 PRINT"ment.  It sends X (a random number) and the number 4, ANY"
```

```
7390 PRINT"valid variable can be passed to a function, even a string."
7400 PRINT
7410 INPUT"press ENTER";T$
7420 GOSUB 21000
7430 GOSUB 44000
7440 PRINT
7450 PRINT"A string could be passed in a function, but it would have to"
7460 PRINT"be operated on legally within the DEF statement.  In this case"
7470 PRINT"the DEF statement in line 20 expects two numerical variables."
7480 PRINT"Note that the two variables passed in line 50 do not match"
7490 PRINT"the DEFined variables.  This is also legal.  You can view the"
7500 PRINT"DEF statement as its own little program, it does not know the"
7510 PRINT"value of any of the variables in the outside program."
7520 INPUT"press ENTER";T$
7530 GOSUB 21000
7540 GOSUB 44000
7550 PRINT
7560 PRINT"The two values that are DEFined as A, and B will be set equal"
7570 PRINT"to the corresponding values of 1 and 4.  A will equal 1 and B"
7580 PRINT"will equal 4.  The DEF statement will then use these numbers to"
7590 PRINT"calculate the (exp) part of the statement, and then Y will be"
7600 PRINT"made equal to this value.  Finally, the value will be printed"
7610 PRINT"in line 50.  You may wish to copy a program like this and "
7620 PRINT"experiment with it to see how it works."
7630 INPUT"press ENTER";T$
7631 GOSUB 21000
7632 GOSUB 44000
7633 PRINT
7634 PRINT"What value would be passed to B in the second DEF statement?"
7635 PRINT
7636 PRINT"A  4"
7637 PRINT"B  The value of X"
7638 PRINT"C  Cannot tell - not enough information"
7639 PRINT
7640 INPUT"ENTER the letter opposite the correct answer";T$
7641 PRINT
7642 IF T$ = "A" THEN GOTO 7646
7643 PRINT"WRONG - the correct answer is A"
7644 GOTO 7647
7645 PRINT"CORRECT"
7647 PRINT
7648 INPUT"press ENTER";T$
7649 GOSUB 21000
7650 GOSUB 45000
7660 PRINT
7670 PRINT"Here is an example of passing a string in a user function. "
```

```
7680 PRINT"In all the examples that we have shown, we have always told"
7690 PRINT"the computer when we wanted to invoke a user function by using"
7700 PRINT"the prefix 'FN(var)'. That is the only way to get to your "
7710 PRINT"user function."
7720 PRINT
7722 INPUT"press ENTER";T$
7724 GOSUB 21000
7726 GOSUB 45000
7728 PRINT
7730 PRINT"Note in the example above, that we passed a string within a"
7740 PRINT"numeric variable, but that in the DEF statement, we set the"
7750 PRINT"numeric variable equal to LEN(string).  LEN(string) is an'
7760 PRINT"example of yet another type of FUNCTION that we will study in"
7770 PRINT"the next lesson.  LEN(string) returns a numeric value equal to"
7772 PRINT"the LENGTH of the string that is within the parenthesis."
7780 PRINT"Don't worry about it now, we will cover it in the next lesson."
7782 PRINT"However, because it is a NUMERIC value, its type matches the"
7784 PRINT"DEF statement, and therefore it is valid."
7790 PRINT
7800 INPUT"press ENTER";T$
7810 GOSUB 21000
7820 PRINT"What kind of function do you create with the DEF statement"
7830 PRINT
7840 PRINT"A  USER"
7850 PRINT"B  LIBRARY"
7860 PRINT"C  EITHER USER or LIBRARY"
7870 PRINT"D  DEFINED FUNCTIONS"
7880 PRINT
7890 INPUT"ENTER the letter opposite the correct answer";T$
7900 PRINT
7910 IF T$ = "A" THEN GOTO 7940
7920 PRINT"WRONG - only USER defined functions are created with the DEF'
7925 PRINT"        statement.  The correct answer is A"
7930 GOTO 7950
7940 PRINT"CORRECT ' - but that was an easy one, try the next question if"
7945 PRINT"           you think you are up to it'"
7946 PRINT"                 (we know you are - we're just trying"
7947 PRINT"                  to lighten things up a bit)"
7950 PRINT
7960 INPUT"press ENTER";T$
7970 GOSUB 21000
7980 PRINT"Is the following statement valid?"
7981 PRINT
7982 PRINT"10 DEF FN88(A,B) = A * B"
7983 PRINT
7984 PRINT"A  Yes, to call on it simply put FN88 somewhere in your program"
```

```
 7985 PRINT"B  No, the teras A and B are not defined and will cause errors"
 7986 PRINT"C  No, 88 is not a valid variable for this position"
 7987 PRINT"D  No, the syntax is good, but the beat is poor, you can't "
 7988 PRINT"       dance to it."
 7989 PRINT
 7990 INPUT"ENTER the letter opposite the correct answer";T$
 7991 PRINT
 7992 IF T$ = "D" THEN PRINT"OK wise guy, even my 8 year old knew that was":PRINT"a stupid answer.  Try
      again.":PRINT:INPUT"press ENTER";T$:GOSUB 21000:GOTO 7980
 7993 IF T$ = "C" THEN GOTO 7996
 7994 PRINT"WRONG - the correct answer is C"
 7995 GOTO 7997
 7996 PRINT"CORRECT"
 7997 PRINT
 7998 INPUT"press ENTER";T$
 7999 GOSUB 40000
 8000 IF T$ = "B" THEN GOTO 7000
 8010 RETURN
 8020 GOSUB 21000
 8030 PRINT"You have completed this lesson and now you can go to the test"
 8040 PRINT"If you wish to review parts of this lesson, type in an 'R'."
 8050 PRINT"If you wish to continue to the test, type in a 'C'."
 8060 PRINT
 8070 INPUT"Enter your choice (R or C)";T$
 8080 PRINT
 8090 IF T$ = "R" THEN RUN
 8100 IF T$ = "C" THEN GOTO 47000
 8110 GOTO 8020
20980 REM **
20990 REM ** This subroutine clears the screen on any terminal
20995 REM **
21000 FOR X = 1 TO 24
21010    PRINT
21020 NEXT X
21030 RETURN
30000 PRINT"                    LESSON 5A"
30010 PRINT
30015 PRINT"This is the second part of a two part lesson"
30020 PRINT"It is divided into the following sections."
30025 PRINT
30030 PRINT"1)  Introduction          4)  User Defined Functions"
30035 PRINT"2)  Functions Overview    5)  DEF Statements"
30040 PRINT"3)  Library Functions     6)  TEST"
30050 PRINT
30090 PRINT
30100 RETURN
```

```
40000 GOSUB 21000
40005 PRINT"Which do you wish to do?"
40010 PRINT
40020 PRINT"A  Continue on"
40030 PRINT"B  Review this section again"
40040 PRINT
40050 INPUT"press the letter opposite the correct answer and press ENTER";T$
40060 IF T$ = "A" OR T$ = "B" THEN RETURN
40070 GOTO 40000
40970 REM **
40980 REM ** these are for Library functions examples
40990 REM **
41000 PRINT"1) ABS(exp) - gives absolute value  7) LOG(exp) - gives LOG(e)"
41010 PRINT"2) ATN(exp) - arctangent in radians 8) RND(0) - random numbers"
41020 PRINT"3) COS(exp) - returns cosine of exp 9) SGN(exp) - sign of exp"
41030 PRINT"4) EXP(exp) - natural exponential   10) SIN(exp) - sine of exp"
41040 PRINT"5) FIX(exp) - gives integer of exp  11) SQR(exp) - square root"
41050 PRINT"6) INT(exp) - gives integer of exp  12) TAN(exp) - tangent(exp)"
41060 PRINT
41070 PRINT"exp = any appropriate numeric expression - remember, you"
41080 PRINT"can't use ANY number for some of the functions (SQR(-1) bombs!)
41090 RETURN
41970 REM **
41980 REM ** the subroutine is for the library function area
41990 REM **
42000 PRINT"10 INPUT"CHR$(34)"Type in the number you want the square root of"CHR$(34)";N"
42010 PRINT"20 PRINT SQR(N)"
42020 PRINT"30 GOTO 10"
42030 RETURN
42970 REM **
42980 REM ** this is another example for the Library functions
42990 REM **
43000 PRINT"10 INPUT"CHR$(34)"Enter the number you want the square root of"CHR$(34)";N"
43010 PRINT"20 PRINT SQR(ABS(N))"
43020 PRINT"30 GOTO 10"
43030 RETURN
43970 REM **
43980 REM ** this example is for user functions
43990 REM **
44000 PRINT"10 DEF FNR2 = INT(RND(0) * 10)"
44010 PRINT"20 DEF FNW(A,B) = A * B/2 + (A - B)"
44020 PRINT"30 X = FNR2"
44030 PRINT"40 PRINT X"
44040 PRINT"50 Y = FNW(X,4)"
44045 PRINT"60 PRINT Y"
44050 PRINT"70 END"
```

```
44060 RETURN
44970 REM **
44980 REM ** example of DEF with strings
44990 REM **
45000 PRINT"10 DEF FNX(A$) = LEN(A$)"
45010 PRINT"20 A$ = "CHR$(34)"This is a function"CHR$(34)
45020 PRINT"30 X = FNX(A$)"
45030 PRINT"40 PRINT X"
45040 RETURN
47000 GOSUB 21000
47010 PRINT"Going to test number 5 - wait patiently please"
47020 RUN "TEST5"
48000 RUN "MENU"
49000 RUN"LESSON5"
50000 END
```

```
1000 REM **
1010 REM ** LESSON:  TEST5                VERSION: 1 AUG 83
1020 REM ** AUTHOR:  CAPT DAN CREAGAN
1030 REM **          AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM **            N$(X) = NAMES ARRAY, USED TO READ IN SEQ-
1070 REM **                    UENTIAL NAMES, AND TO WRITE OUT
1080 REM **                    UPDATE NAMES.
1090 REM **            S(X)  = SCORES ARRAY - USED TO READ AND
1100 REM **                    WRITE SCORES
1110 REM **            Q(X)  = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM **                    CORRECT ANSWERS.  IF AN ARRAY
1130 REM **                    ELEMENT EQUALS 1, THE ANSWER WAS
1140 REM **                    CORRECT
1150 REM **
1160 CLEAR 3000
1170 GOSUB 4130
1180 DIM N$(1000)
1190 DIM Q(10)
1200 DIM S(1000)
1210 PRINT"                    FINAL TEST (lesson 5)"
1220 PRINT
1230 PRINT"This test consists of 10 questions, you must get 70 percent"
1240 PRINT"of them correct to pass.  (that's 7 right out of the 10 ques-"
1250 PRINT"tions).  Use only capital letters in your answers, don't"
1260 PRINT"include extra spaces or letters.  If you successfully complete"
1270 PRINT"the test, you can go on to the last lesson!"
1280 PRINT"                      GOOD LUCK"
1290 PRINT
1300 INPUT"press ENTER to continue";T$
1310 GOSUB 4130
1320 PRINT"Type in the order in which the following lines will be executed"
1330 PRINT"Leave one space between each line number - ie, if the execution"
1340 PRINT"sequence is ten, twenty and thirty, then type in 10 20 30"
1350 PRINT
1360 PRINT"10 GOSUB 1000"
1370 PRINT"20 PRINT "CHR$(34)"DONE"CHR$(34)
1380 PRINT"30 END"
1390 PRINT"1000 RETURN"
1400 PRINT
1410 INPUT"Enter the sequence now";T$
1420 PRINT
1430 IF T$ = "10 1000 20 30" THEN GOTO 1470
1440 PRINT"WRONG - the correct answer is 10 1000 20 30"
1450 PRINT"       See part 1, Subroutines"
```

```
1460 GOTO 1490
1470 PRINT"CORRECT"
1480 Q(1) = 1
1490 PRINT
1500 INPUT"press ENTER";T$
1510 GOSUB 4130
1520 PRINT"What will happen when you input then number 4 to the following"
1530 PRINT"program?"
1540 PRINT
1550 PRINT"10 INPUT A"
1560 PRINT"20 ON A GOSUB 1000,2000,3000"
1570 PRINT"30 END"
1580 PRINT".... rest of program is not important"
1590 PRINT
1600 PRINT"A  Nothing"
1610 PRINT"B  Subroutine 3000 would be called"
1620 PRINT"C  Subroutine 1000 would be called"
1630 PRINT"D  The program would end"
1640 PRINT
1650 GOSUB 4200
1660 PRINT
1670 IF T$ = "D" THEN GOTO 1720
1680 PRINT"WRONG - the correct answer is D"
1690 PRINT"       If you don't satisfy one of the ON GOSUB choices"
1700 PRINT"       the line defaults to the line just below it."
1710 PRINT"       See part 1. ON GOSUB"
1720 GOTO 1750
1730 PRINT"CORRECT"
1740 Q(2) = 1
1750 PRINT
1760 INPUT"press ENTER";T$
1770 GOSUB 4130
1780 PRINT"How many RETURNS can you have in a subroutine?"
1790 PRINT
1800 PRINT"A  One"
1810 PRINT"B  As many as you want, but they should be kept to a minimum"
1820 PRINT"C  One for every GOSUB"
1830 PRINT"D  No more than the amount of memory available"
1840 PRINT
1850 GOSUB 4200
1860 PRINT
1870 IF T$ = "B" THEN GOTO 1910
1880 PRINT"WRONG - the correct answer is B"
1890 PRINT"       See part 1. Subroutines"
1900 GOTO 1930
1910 PRINT"CORRECT"
```

```
1920 Q.5) = 1
1930 PRINT
1940 INPUT"press ENTER";T$
1950 GOSUB 4130
1960 PRINT"What is wrong with this program?"
1970 PRINT
1980 PRINT"10 INPUT A"
1990 PRINT"20 GOSUB 1000"
2000 PRINT"30 END"
2010 PRINT"1000 IF A .= 10 THEN RETURN"
2020 PRINT"10005 IF A = 11 THEN RETURN"
2030 PRINT"1010 GOTO 10"
2040 PRINT
2050 PRINT"A  Nothing"
2060 PRINT"B  One of the possible exits from the subroutine is incorrect"
2070 PRINT"C  There are too many RETURN statements"
2080 PRINT"D  The IF statements aren't allowed in a subroutine like this."
2090 PRINT
2100 GOSUB 4200
2110 PRINT
2120 IF T$ = "B" THEN GOTO 2190
2130 PRINT"WRONG the correct answer is B"
2140 PRINT"      If A is greater than 10, then line 1010 causes"
2150 PRINT"      the subroutine to branch into the main program."
2160 PRINT"      That will eventually cause the computer to get"
2170 PRINT"      mixed up and BOMB.  See part 1, Subroutines."
2180 GOTO 2210
2190 PRINT"CORRECT"
2200 Q.5) = 1
2210 PRINT
2220 INPUT"press ENTER";T$
2230 GOSUB 4130
2240 PRINT"For the next few questions, you should be sure you have your"
2250 PRINT"BASIC manual available so you can figure them out correctly."
2260 PRINT"All the questions are from part 2 of the lesson, and your"
2270 PRINT"BASIC manual."
2280 PRINT
2290 INPUT"press ENTER";T$
2300 GOSUB 4130
2310 PRINT"What is the value of the following statement"
2320 PRINT
2330 PRINT"SQR 25"
2340 PRINT
2350 PRINT"A  5
2360 PRINT"B  625"
2370 PRINT"C  the natural logarithm of 25"
```

```
***** Listing of Program 'TEST5' *****                    07/10.81 - 04:02:56

2280 PRINT"D  100"
2290 PRINT
2400 GOSUB 4200
2410 PRINT
2420 IF T$ = "A" THEN GOTO 2450
2430 PRINT"WRONG - the correct answer is A"
2440 GOTO 2470
2450 PRINT"CORRECT"
2460 D(5) = 1
2470 PRINT
2480 INPUT"press ENTER";T$
2490 GOSUB 4100
2500 PRINT"what type of functions would be written by you"
2510 PRINT
2520 PRINT"type in your answer using all capital letters. Do NOT append"
2530 PRINT"the word FUNCTION on the end of your answer."
2540 PRINT
2550 INPUT"type in your answer now";T$
2560 PRINT
2570 IF T$ = "USER" THEN GOTO 2600
2580 PRINT"WRONG - the correct answer is USER"
2590 GOTO 2620
2600 PRINT"CORRECT"
2610 D(6) = 1
2620 PRINT
2630 INPUT"press ENTER";T$
2640 GOSUB 4100
2650 PRINT"what is the term you would use to define a USER function."
2660 PRINT"hint: it's two words. DON'T include a variable, leave a space"
2670 PRINT"between the two words!
2680 PRINT
2690 INPUT"ENTER your answer now";T$
2700 PRINT
2710 IF T$ = "DEF FN" THEN GOTO 2740
2720 PRINT"WRONG - the correct answer is DEF FN"
2730 GOTO 2760
2740 PRINT"CORRECT"
2750 D(7) = 1
2760 PRINT
2770 INPUT"press ENTER";T$
2780 GOSUB 4100
2790 PRINT"10 DEF FNX = 10 * 2"
2800 PRINT"20 Y = FNX"
2810 PRINT
2820 PRINT"what is the term that would complete line 20 if we wished to"
2830 PRINT"call on the user function in line 10"
```

```
 2540 PRINT
 2550 INPUT"ENTER your answer now";T$
 2560 PRINT
 2570 IF T$ = "FNx" THEN GOTO 2600
 2580 PRINT "WRONG - the correct answer would be FNX"
 2590 GOTO 2620
 2600 PRINT"CORRECT"
 2610 Q(5) = 1
 2620 PRINT
 2630 INPUT"press ENTER";T$
 2640 GOSUB 4100
 2650 PRINT"Which of the following functions is valid"
 2660 PRINT
 2670 PRINT"A  DEF FNx(X,M) = 2 * M 2 + A$"
 2680 PRINT"B  DEF FN10(2,M) = 2 * 2 - 2 + M"
 2690 PRINT"C  DEF FNx(A$,X) = LEN A$) + X"
 2700 PRINT"D  DEF FNx = SQR(-4)"
 2710 PRINT"E  DEF FNx = LOG(0)"
 2720 PRINT
 2730 GOSUB 4200
 2740 PRINT
 2750 IF T$ = "C" THEN GOTO 2780
 2760 PRINT "WRONG - the correct answer is C
 2770 GOTO 2810
 2780 PRINT"CORRECT"
 2790 Q(5) = 1
 2800 PRINT
 2810 INPUT"press ENTER";T$
 2820 GOSUB 4100
 2830 PRINT What is the proper statement to exit from a subroutine"
 2840 PRINT
 2850 INPUT"ENTER your answer now ";T$
 2860 PRINT
 2870 IF T$ = "RETURN" THEN GOTO 2910
 2880 PRINT"WRONG - the correct answer is RETURN"
 2890 PRINT"        See part 1, Subroutines"
 2900 GOTO 2930
 2910 PRINT"CORRECT"
 2920 Q(10) = 1
 2930 PRINT
 2940 INPUT"press ENTER";T$
 2950 GOSUB 4100
 2960 FOR X = 1 TO 10
 2970     Y = Y+Q(X
 2980 NEXT X
 2990 PRINT "You have finished the test. out of 10 possible correct answers"
```

```
3200 PRINT"You scored "Y"."
3210 PRINT
3220 IF Y > 6 THEN  PRINT"YOU HAVE PASSED"
3230 GOSUB 3990
3240 IF Y > 6 THEN GOTO 3420
3250 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
3260 PRINT
3270 PRINT"YOU SHOULD RETAKE LESSON 5!"
3280 PRINT
3290 PRINT"You will be returned to the Menu."
3400 PRINT
3410 GOTO 4220
3420 PRINT
3430 PRINT"Do you want your score recorded on a permanent file?"
3440 PRINT
3450 PRINT"A  YES"
3460 PRINT"B  NO"
3470 PRINT
3480 INPUT"Which";T$
3490 IF T$ = "B" THEN GOTO 3790
3500 GOSUB 4170
3510 PRINT"To record your score, we must open a file and put your name"
3520 PRINT"in it.  Therefore, surprisingly, we need your name.  If your"
3530 PRINT"name is not unique among the students likely to take this test,"
3540 PRINT"please contact your test monitor for an identifying word that"
3550 PRINT"will make you unique.  Then enter that word below."
3560 PRINT
3570 PRINT"If you have already entered a score previously, be sure to"
3580 PRINT"enter the same name you used before.  Use all capitals."
3590 PRINT
3600 INPUT"ENTER your word or name now";T$
3610 OPEN"I",1,"SCORES"
3620 X = 0
3630 IF EOF(1) THEN GOTO 3690
3640 X = X+1
3650 INPUT#1,N$(X)
3660 INPUT#1,S(X)
3670 IF N$(X) = T$ THEN GOTO 3730
3680 GOTO 3630
3690 CLOSE
3700 X = X+1
3710 N$(X) = T$
3720 S(X) = Y
3730 OPEN"O",1,"SCORES"
3740 FOR W = 1 TO X
3750   PRINT#1,N$(W)
```

```
3760    PRINT@1,S(W)
3770 NEXT W
3780 PRINT
3790 GOSUB 4130
3800 PRINT"You are now qualified to go to LESSON 6."
3810 PRINT"You may return to the MENU or receive your homework."
3820 PRINT
3830 INPUT"Do you want your homework assignment (Y/N)";T$
3840 IF T$ = "Y" THEN GOTO 4240
3850 PRINT
3860 GOTO 4220
3870 S X$ = "
3880 IF EOF(1) THEN CLOSE:GOTO 3920
3890 X = X+1
3900 INPUT#1, N$(X), S(X)
3910 GOTO 3880
3920 OPEN"O",1,"SCORES"
3930    FOR W = 1 TO X
3940        PRINT#1,N$(W)
3950        PRINT#1,S(W)
3960    NEXT W
3970 PRINT
3980 GOTO 3300
3990 IF <=10 THEN RETURN
4000 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
4010 PRINT
4020 IF Q(1) = 0 OR Q(3) = 0 OR Q(4) = 0 OR Q(10) = 0 THEN PRINT"    part 1. SUBROUTINES"
4030 IF Q(2) = 0 THEN PRINT"    part 1. ON GOSUB"
4040 IF Q(5) = 0 OR Q(6) = 0 OR Q(7) = 0 THEN PRINT"    part 2. Library Functions"
4050 IF Q(8) = 0 OR Q(9) = 0 THEN PRINT"    part 2. USER Functions"
4060 PRINT
4070 INPUT"press ENTER";T$
4080 GOSUB 4130
4090 RETURN
4100 REM **
4110 REM ** this    routine clears the screen"
4120 REM **
4130 FOR X = 1 TO 24
4140     PRINT
4150 NEXT X
4160 RETURN
4170 REM **
4180 REM ** this subroutine is for the response section
4190 REM
4200 INPUT"ENTER the letter opposite the correct answer";T$
4210 RETURN
```

```
4220 PRINT
4230 RUN"MENU"
4240 RUN"HW5"
4250 CLOSE
4260 OPEN"I",1,"SCORE5"
4270  IF EOF(1) THEN STOP
4280 INPUT#1,A$,N
4290 PRINTA$,N
4300 GOTO 4270
```

```
***** Listing of Program 'LESSON5' *****                    07/10/87 - 04:14:15

 1000 REM ** THIS PROGRAM STARTED ON 17 JUNE 1987
 1010 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
 1020 REM ** TITLE: LESSON 5
 1030 REM **
 1040 REM **
 1050 REM **
 1060 REM **
 1070 CLEAR 200
 1080 GOSUB 6510
 1090 PRINT"LESSON:  BASIC 5            VERSION: 1 AUGUST 87
 1100 PRINT
 1110 PRINT"TIME REQUIRED TO COMPLETE LESSON:  Less than one hour"
 1120 PRINT
 1130 PRINT
 1140 PRINT"AUTHOR:  Capt Danny J. Creagan"
 1150 PRINT"         Air Force Institute of Technology"
 1160 PRINT
 1170 PRINT"OBJECTIVE:  To teach the student about string functions "
 1180 PRINT"            and the Microsoft Editor"
 1190 PRINT
 1200 PRINT
 1210 PRINT
 1220 PRINT
 1230 PRINT
 1240 INPUT"press the ENTER key to continue";T$
 1250 GOSUB 6510
 1260 GOSUB 6550
 1270 PRINT"A  I'm taking this part in its entiret.."
 1280 PRINT"B  I wish to review selected areas."
 1290 PRINT"C  I want to go to the second part."
 1300 PRINT"D  I want to return to the Menu."
 1310 PRINT
 1320 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
 1330 IF T$ = "D" GOTO 7020
 1340 IF T$ = "C" GOTO 7030
 1350 IF T$ = "B" GOTO 1450
 1360 IF T$ <>"A" GOTO 1270
 1370 GOSUB 1560
 1380 GOSUB 2220
 1390 GOSUB 3220
 1400 GOSUB 4090
 1410 GOSUB 4560
 1420 GOSUB 6390
 1430 GOSUB 6510
 1440 GOTO 7000
 1450 GOSUB 6510
```

```
1460 GOSUB 6550
1470 PRINT
1480 PRINT"Please type in the number beside the area you wish"
1490 PRINT"to review (1 through 5) and then press ENTER - press 0 and"
1500 PRINT"press ENTER to return to the Menu."
1510 PRINT
1520 INPUT"What is your choice";N
1530 IF N = 0 GOTO 7020
1540 ON N GOSUB 1560,2220,3220,4090,4500
1550 GOTO 1450
1560 GOSUB 6510
1570 PRINT"                                INTRODUCTION"
1580 PRINT
1590 PRINT"This is your last lesson! Congratulations! If you have taken"
1600 PRINT"the previous five lessons, you should be feeling a little more"
1610 PRINT"comfortable with Microsoft BASIC by now."
1620 PRINT
1630 PRINT"In this lesson we will cover STRING functions first, and then"
1640 PRINT"we will learn about the Microsoft Editor (in the second half)."
1650 PRINT
1660 PRINT"As we learned before, a BASIC string is one or more alpha-"
1670 PRINT"numeric characters that are treated as a single collection of"
1680 PRINT"data.  Using the concepts in this chapter, you can perform"
1690 PRINT"many of the same types of operations on STRINGs that you "
1700 PRINT"can perform on numeric data."
1710 PRINT
1720 INPUT"press ENTER";I$
1730 GOSUB 6510
1740 PRINT"                           INTRODUCTION"
1750 PRINT
1760 PRINT"As a small review, you should remember that string data can be"
1770 PRINT"designated in two ways.  You can assign your data to a string"
1780 PRINT"variable, or you can enclose the data in quotes.  Here are two"
1790 PRINT"examples."
1800 PRINT
1810 PRINT"10 PRINT "CHR$(34)"This is one way to designate a string "CHR$(34)
1820 PRINT"20 V$ = "CHR$(34)"Another way is to put it in a variable and print it"CHR$(34)
1830 PRINT"30 PRINT V$"
1840 PRINT
1850 PRINT"Line 10 prints the string data immediately, and line 20 loads"
1860 PRINT"the variable V$ with the data.  V$ can then be printed when-"
1870 PRINT"ever we want it."
1880 PRINT
1890 INPUT"press ENTER";I$
1900 GOSUB 6510
1910 PRINT"                               INTRODUCTION"
```

```
1920 PRINT
1930 PRINT"10 PRINT "CHR$(34)"This is one way to designate a string"CHR$(34)
1940 PRINT"20 V$ = "CHR$(34)"And this is another"CHR$(34)
1950 PRINT"30 PRINT"
1960 PRINT"40 PRINT V$"
1970 PRINT"RUN"
1980 PRINT
1990 PRINT"This is one way to designate a string"
2000 PRINT
2010 PRINT"And this is another"
2020 PRINT
2030 PRINT"Here is another variation of our little program.  Note that"
2040 PRINT"'And this is another' is not printed until line 40 is executed."
2050 PRINT
2060 INPUT"press ENTER";T$
2070 GOSUB 6510
2080 PRINT"                              INTRODUCTION"
2090 PRINT
2100 PRINT"Remember, in this lesson, as in all of our lessons, you should"
2110 PRINT"have either a good BASIC manual handy, or you should have"
2120 PRINT"an experienced programmer around to help you with difficult"
2130 PRINT"problems. "
2140 PRINT
2150 PRINT"In some of the answers you will need to be sure you use the"
2160 PRINT"correct case (either uppercase or lowercase), so be sure to"
2170 PRINT"read all the questions carefully."
2180 PRINT
2190 PRINT"Get out your manual, or programmer, now, and let's enjoy BASIC!"
2200 PRINT
2210 INPUT"press ENTER";T$
2220 GOSUB 6510
2230 PRINT"                        STRING ASSIGNMENT"
2240 PRINT
2250 PRINT"As we showed you in the introduction, you assign strings to "
2260 PRINT"a variable and then you can print the variable anywhere in the"
2270 PRINT"program.  That makes it easier to write long program lines,"
2280 PRINT"because you don't have to keep typing in the text every time"
2290 PRINT"you want to use the string data."
2300 PRINT
2310 PRINT"You can assign data to strings using any of the statements we"
2320 PRINT"used to assign numeric data to numeric variables.  LET, READ,"
2330 PRINT"and INPUT are all used with string assignment (LET is optional"
2340 PRINT"just as it is with numeric data)."
2350 PRINT
2360 INPUT"press ENTER";T$
2370 GOSUB 6510
```

```
2380 PRINT"Are these statements legal? (assume the program is just for"
2390 PRINT"demonstration, and that B$ is blank)"
2400 PRINT
2410 PRINT"10 READ A$"
2420 PRINT"20 INPUT A$"
2430 PRINT"30 LET A$ = B$
2440 PRINT"40 A$ = "CHR$(34)"NOW IS THE TIME"CHR$(34)
2450 PRINT"50 DATA "CHR$(34)"NOW IS THE TIME"CHR$(34)
2460 PRINT
2470 PRINT"A  No, the LET statement in line 30 is illegal"
2480 PRINT"B  No, the string assignment in line 40 is illegal"
2490 PRINT"C  No, you cannot read data into a string (line 10 is bad)"
2500 PRINT"D  Yes, all statements are legal"
2510 PRINT
2520 INPUT"ENTER the letter opposite the correct answer";T$
2530 PRINT
2540 IF T$ = "D" THEN GOTO 2570
2550 PRINT"WRONG - all these assignments are legal"
2560 GOTO 2530
2570 PRINT"CORRECT - SUPER "
2580 PRINT
2590 A$ = "10  NN$ = "+CHR$(34)+" NOW IS THE TIME "+CHR$(34)
2600 INPUT"press ENTER";T$
2610 GOSUB 6510
2620 PRINT"                    String Assignment"
2630 PRINT
2640 GOSUB 6730
2650 PRINT"RUN"
2660 PRINT
2670 PRINT"THE GRINCH"
2680 PRINT"IS COMING"
2690 PRINT
2700 PRINT"Notice that A$ was converted to B$, and all the data was"
2710 PRINT"printed out by using just A$ in print statements."
2720 PRINT
2730 INPUT"press ENTER";T$
2740 GOSUB 6510
2750 PRINT"                    String Assignment"
2760 PRINT
2770 GOSUB 6730
2780 PRINT
2790 PRINT"Notice the dollar sign is always included with a string "
2800 PRINT"variable.  The dollar sign tells the computer to treat the"
2810 PRINT"variable as a string instead of as a numeric.  Also, whenever"
2820 PRINT"you assign data to a string, it must either be another string"
2830 PRINT"or it must be enclosed in quotes."
```

```
2940 PRINT
2950 INPUT"press ENTER";T$
2960 GOSUB 6510
2970 PRINT"Assign NOW IS THE TIME to a string variable called NN$"
2980 PRINT"and use line number 10 as your statement number."
2990 PRINT
3000 PRINT"Put one blank between terms."
3010 PRINT
3020 LINE INPUT"ENTER your answer now ? ";T$
3030 G$ = "10 NN$ = "+CHR$(34)+"NOW IS THE TIME"+CHR$(34)
3040 H$ = "10 NN$ = "+CHR$(34)+" NOW IS THE TIME "+CHR$(34)
3050 PRINT
3060 IF T$ = G$ OR T$ = H$  THEN GOTO 3090
3070 PRINT"WRONG - the correct answer is 10 NN$ = "CHR$(34)"NOW IS THE TIME"CHR$(34)
3080 PRINT"     (you could have also answered 10 NN$ = "CHR$(34)" NOW IS THE TIME "CHR$(34)")"
3090 GOTO 3110
3100 PRINT"CORRECT"
3110 PRINT
3120 INPUT"press ENTER";T$
3130 GOSUB 6510
3140 PRINT"You can also INPUT$ string data from an external file."
3150 PRINT"(the following program assumes that a file named TEST was"
3160 PRINT"previously created on disk)"
3170 PRINT
3180 PRINT"10 OPEN"CHR$(34)"I"CHR$(34)",1,"CHR$(34)"TEST"CHR$(34)
3190 PRINT"20 IF EOF(1) THEN END"
3200 PRINT"30 INPUT#1,A$"
3210 PRINT"40 PRINT A$"
3220 PRINT"50 GOTO 20"
3230 PRINT
3240 PRINT"A program such as this is used to read in your name when you"
3250 PRINT"take your test at the end of each lesson.  The original is"
3260 PRINT"enhanced a little, but the BASIC idea is the same."
3270 PRINT
3280 INPUT"press ENTER";T$
3290 GOSUB 6660
3300 IF T$ = "B" THEN GOTO 2220
3310 RETURN
3320 GOSUB 6510
3330 PRINT"                          String ARRAYs"
3340 PRINT
3350 PRINT"You can assign string data to arrays in the same way as you"
3360 PRINT"assign numeric data to arrays.  Nearly all the rules are the"
3370 PRINT"same.  The following is an example."
3380 PRINT
3390 GOSUB 6870
```

```
3300 PRINT
3310 INPUT"press ENTER";T$
3320 GOSUB 6510
3330 GOSUB 6870
3340 PRINT
3350 PRINT"Notice the CLEAR statement.  Remember that you normally have"
3360 PRINT"only 50 - 100 characters of string space available, and if you"
3370 PRINT"are going to need more, you need to tell the computer.  Also,"
3380 PRINT"note the DIM statement - we need declare our array size if it "
3390 PRINT"is over 10"
3400 PRINT
3410 INPUT"press ENTER";T$
3420 GOSUB 6510
3430 GOSUB 6870
3440 PRINT
3450 PRINT"The variable X acts as a counter to reference the proper pocket of"
3460 PRINT"the string array.  The string array is referenced exactly like"
3470 PRINT"the numeric array.  Note that this program will only read in"
3480 PRINT"the data.  If you want to print it out, you will have to add"
3490 PRINT"some more statements on the bottom of the program."
3500 PRINT
3510 INPUT"press ENTER";T$
3520 GOSUB 6510
3530 PRINT"What is the CLEAR statement for in BASIC?"
3540 PRINT
3550 PRINT"A  To clear extra number space for the computer"
3560 PRINT"B  To clear extra string space"
3570 PRINT"C  To zeroize all number variables"
3580 PRINT"D  To clear the screen"
3590 PRINT"E  To help the programmer understand more clearly"
3600 PRINT
3610 INPUT"ENTER the letter opposite the correct answer";T$
3620 PRINT
3630 IF T$ = "B" THEN GOTO 3680
3640 PRINT"WRONG - the correct answer is B"
3650 PRINT"       this is an important concept, you may wish to review"
3660 PRINT"       lesson 3 before you go to the next section."
3670 GOTO 3690
3680 PRINT"CORRECT"
3690 PRINT
3700 INPUT"press ENTER";T$
3710 GOSUB 6510
3720 PRINT"                         String ARRAYs"
3730 PRINT
3740 GOSUB 6870
3750 PRINT
```

```
3760 PRINT"The rule for the DIM statement is the same as for numeric"
3770 PRINT"arrays.  What is the maximum size of one leg of a string array"
3780 INPUT"if you don't use the DIM statement ... type your answer now";T$
3790 PRINT
3800 IF T$ = "10" THEN GOTO 3850
3810 PRINT"WRONG - the max size of an array without a DIM statement is 10"
3820 PRINT"       this is an important concept, you may wish to review"
3830 PRINT"       lesson 3 before you go to the next section."
3840 GOTO 3860
3850 PRINT"CORRECT"
3860 PRINT
3870 INPUT"press ENTER";T$
3880 GOSUB 6510
3890 PRINT"How would you find out what was in the fifth pocket of the"
3900 PRINT"single dimension array A$(X)?"
3910 PRINT
3920 PRINT"A  PRINT A$(X)"
3930 PRINT"B  PRINT A$(5)"
3940 PRINT"C  PRINT A$"
3950 PRINT"D  READ A$(X)"
3960 PRINT
3970 INPUT"ENTER the letter opposite the correct answer";T$
3980 PRINT
3990 IF T$ = "B" THEN GOTO 4040
4000 PRINT"WRONG - the correct answer is B"
4010 PRINT"       this is an important concept, you may wish to review"
4020 PRINT"       lesson 3 before you go to the next section."
4030 GOTO 4050
4040 PRINT"CORRECT"
4050 PRINT
4060 INPUT"press ENTER";T$
4065 GOSUB 6560
4070 IF T$ = "S" THEN GOTO 3220
4080 RETURN
4090 GOSUB 6510
4100 PRINT"                    Concatenation"
4110 PRINT
4120 PRINT"You may link two strings together by using the 'plus' symbol."
4130 PRINT"For example:"
4140 PRINT
4150 PRINT"10 A$ = "CHR$(34)"where"CHR$(34)
4160 PRINT"20 B$ = "CHR$(34)"Some"CHR$(34)
4170 PRINT"30 C$ = B$ + A$"
4180 PRINT"40 PRINT C$  :'you could have said 'PRINT B$ + A$' too."
4190 PRINT"RUN"
4200 PRINT
```

```
4210 PRINT"Somewhere"
4220 PRINT
4230 PRINT"In this case, the '+' symbol serves to 'add' the two strings"
4240 PRINT"together and create another string."
4250 INPUT"press ENTER";T$
4260 GOSUB 6510
4270 PRINT"What is the output of the following program?"
4280 PRINT
4290 PRINT"10  A$ = "CHR$(34)"FLASH"CHR$(34)
4300 PRINT"20  B$ = "CHR$(34)"DANCE"CHR$(34)
4310 PRINT"30  PRINT A$ + B$"
4320 PRINT
4330 PRINT"A  FLASH"
4340 PRINT"B  FLASH"
4350 PRINT"   DANCE"
4360 PRINT"C  FLASHDANCE"
4370 PRINT"D  DANCEFLASH"
4380 PRINT
4390 INPUT"ENTER the letter opposite the correct answer";T$
4400 PRINT
4410 IF T$ = "C" THEN GOTO 4440
4420 PRINT"WRONG - the correct answer is C"
4430 GOTO 4450
4440 PRINT"CORRECT"
4450 PRINT
4460 INPUT"press ENTER";T$
4470 GOSUB 6560
4480 IF T$ = "B" THEN GOTO 4090
4490 RETURN
4500 GOSUB 6510
4510 PRINT"                              String Functions"
4520 PRINT
4530 PRINT"For this section you will definitely need your BASIC manual, so"
4540 PRINT"get it out now."
4550 PRINT
4560 PRINT"As with arithmetic functions, there are STRING functions."
4570 PRINT"STRING functions are used to manipulate or explore the contents"
4580 PRINT"of a string.  On the next screen there are several examples of"
4590 PRINT"STRING functions.  We will go over several of these, but you"
4600 PRINT"will not have to memorize them.  Rather, you should understand"
4610 PRINT"that if you need to access or modify any kind of string, you"
4620 PRINT"can probably find a string function that will do the job for"
4630 PRINT"you.  String functions can be used as part of USER functions"
4640 PRINT"as you saw in lesson 5."
4650 PRINT
4660 INPUT"press ENTER for some examples of string functions";T$
```

```
4670 GOSUB 6510
4680 PRINT"                        String Functions"
4690 PRINT
4700 PRINT"1) ASC(string)                  5) LEN(string)"
4710 PRINT"2) CHR$(exp)                    6) MID$(string,position,length)"
4720 PRINT"3) FRE(string)                  7) RIGHT$(string,length)"
4730 PRINT"4) INKEY$                       8) LEFT$(string,length)"
4740 PRINT
4750 PRINT"At first glance, these functions look like a lot of GREEK, in"
4760 PRINT"fact, they look pretty bad at second glance!  However, they"
4770 PRINT"really are pretty easy to use, once you understand them.  The"
4780 PRINT"best way to learn how to use them is to make a short program"
4790 PRINT"and use them one at a time until you see what they do."
4800 PRINT"We will go over examples of a couple to help you catch on."
4810 PRINT
4820 INPUT"press ENTER";T$
4830 GOSUB 6510
4840 PRINT"Is the following statement TRUE or FALSE?"
4850 PRINT
4860 PRINT"String functions are used to manipulate data within string"
4870 PRINT"variables."
4880 PRINT
4890 PRINT"A  TRUE"
4900 PRINT"B  FALSE"
4910 PRINT
4920 INPUT"ENTER the letter opposite the correct answer";T$
4930 PRINT
4940 IF T$ = "A" THEN GOTO 4980
4950 PRINT"WRONG - string functions ARE used to manipulate string"
4960 PRINT"        variables."
4970 GOTO 4990
4980 PRINT"CORRECT"
4990 PRINT
5000 INPUT"press ENTER";T$
5010 GOSUB 6510
5020 PRINT"                    String Functions"
5030 PRINT
5040 GOSUB 6980
5050 PRINT"20 PRINT ASC(A$)"
5060 PRINT
5070 PRINT"ASC(string) is a function that returns the ASCII code of the"
5080 PRINT"first character of the string.  ASCII stands for 'AMERICAN"
5090 PRINT"STANDARD CODE for INFORMATION INTERCHANGE.  Look up the ASCII"
5100 PRINT"code for the first letter of A$ in your BASIC manual.  What is"
5110 PRINT"it?  You should have found it to be 77 decimal."
5120 PRINT"When your computer writes data files to disk, it usually writes"
```

```
5130 PRINT"them in ASCII code, one letter at a time.  This function has"
5140 PRINT"use when you are trying to convert characters to their number"
5150 PRINT"equivalent."
5160 PRINT
5170 INPUT"press ENTER";T$
5180 GOSUB 6510
5190 PRINT"                    String Functions"
5200 PRINT
5210 PRINT"PRINT CHR$(77)"
5220 PRINT
5230 PRINT"CHR$(exp) returns the opposite of the ASC(string) function."
5240 PRINT"It returns a character equivalent of decimal 77.  Which is 'M'."
5250 PRINT
5260 INPUT"press ENTER";T$
5270 GOSUB 6510
5280 PRINT"What is the output of the following program?"
5290 PRINT
5300 PRINT"10 PRINT ASC("CHR$(34)"A"CHR$(34)")"
5310 PRINT"20 PRINT CHR$(65)"
5320 PRINT
5330 PRINT"A  A"
5340 PRINT"   65"
5350 PRINT"B  65"
5360 PRINT"   B"
5370 PRINT"C  A A"
5380 PRINT"D  B"
5390 PRINT"   C"
5400 PRINT
5410 INPUT"ENTER the letter opposite the correct answer";T$
5420 PRINT
5430 IF T$ = "B" THEN GOTO 5480
5440 PRINT"WRONG - the correct answer is 65"
5450 PRINT"                              B"
5460 PRINT"       be sure to use your manual!"
5470 GOTO 5490
5480 PRINT"CORRECT - GREAT!"
5490 PRINT
5500 INPUT"press ENTER";T$
5510 GOSUB 6510
5520 PRINT"                    String Functions"
5530 PRINT
5540 GOSUB 6980
5550 PRINT"20 PRINT LEN(A$)"
5560 PRINT
5570 PRINT"LEN(string) is a function that returns the length of the string"
5580 PRINT"that is in parenthesis.  In this case it should return 17."
```

```
5590 PRINT
5600 PRINT"What is the value of LEN("CHR$(34)"TOM SWIFT"CHR$(34)")?"
5610 PRINT
5620 PRINT"A  11"
5630 PRINT"B  8"
5640 PRINT"C  1"
5650 PRINT"D  9"
5660 PRINT
5670 INPUT"ENTER the letter opposite the correct answer";T$
5680 PRINT
5690 IF T$ = "D" THEN GOTO 5720
5700 PRINT"WRONG - the correct answer is D"
5710 GOTO 5730
5720 PRINT"CORRECT"
5730 PRINT
5740 INPUT"press ENTER";T$
5750 GOSUB 6510
5760 PRINT"Now you have to do some work for yourself.  What will be the"
5770 PRINT"output of the following program?"
5780 PRINT
5790 GOSUB 6980
5800 PRINT"20 PRINT LEFT$(A$,2)"
5810 PRINT
5820 PRINT"A  My"
5830 PRINT"B  M"
5840 PRINT"C  My aching"
5850 PRINT"D  Nothing will be output"
5860 PRINT
5870 INPUT"ENTER the letter opposite the correct answer";T$
5880 PRINT
5890 IF T$ = "A" THEN GOTO 5930
5900 PRINT"WRONG - the correct answer is A"
5910 PRINT
5920 GOTO 5940
5930 PRINT"CORRECT"
5940 PRINT
5950 INPUT"press ENTER";T$
5960 GOSUB 6510
5970 GOSUB 6980
5980 PRINT"20 PRINT MID$(A$,4,6)"
5990 PRINT
6000 INPUT"ENTER the output of this program";T$
6010 PRINT
6020 IF T$ = "aching" THEN GOTO 6050
6030 PRINT"WRONG - the correct answer is aching"
6040 GOTO 6060
```

```
5350 PRINT"CORRECT - I'm glad to see you use the book!"
5360 PRINT
5370 INPUT"press ENTER";T$
5380 GOSUB 6510
5390 GOSUB 6980
5400 PRINT"20 PRINT RIGHT$(A$,4)"
5410 PRINT
5420 INPUT"ENTER the output of this program";T$
5430 PRINT
5440 IF T$ = "gers" THEN GOTO 5470
5450 PRINT"WRONG - the correct answer is gers"
5460 GOTO 5430
5470 PRINT"CORRECT - good job"
5480 PRINT
5490 INPUT"press ENTER";T$
5500 GOSUB 6510
6210 PRINT"                    String Functions"
6220 PRINT
6230 PRINT"Here's an interesting function, it's called INKEY$ and it"
6240 PRINT"strobes your keyboard ONCE and if a key is depressed, it"
6250 PRINT"returns the character that was pressed.  Here is an example of"
6260 PRINT"how to use it."
6270 PRINT
6280 PRINT"10 IF INKEY$ = "CHR$(34)"S"CHR$(34)" THEN END"
6290 PRINT"20 GOTO 10"
6300 PRINT
6310 PRINT"If you type this program in exactly as shown, and RUN it, it "
6320 PRINT"will keep running until you press the 'S' key.  Try it when you"
6330 PRINT"are done here."
6340 PRINT
6350 INPUT"press ENTER";T$
6360 GOSUB 6660
6370 IF T$ = "S" THEN GOTO 4500
6380 RETURN
6390 GOSUB 6510
6400 PRINT"You have finished the first part of lesson 6.  If you wish to"
6410 PRINT"review this part, type in 'R', if you want to continue to the"
6420 PRINT"next half, type in 'C'"
6430 PRINT
6440 INPUT"ENTER an R or a C";T$
6450 IF T$ = "R" THEN RUN
6460 IF T$    "C" THEN GOTO 6390
6470 GOTO 7000
6480 REM **
6490 REM ** This subroutine clears the screen on any terminal
6500 REM **
```

```
6510 FOR X = 1 TO 24
6520    PRINT
6530 NEXT X
6540 RETURN
6550 PRINT"                    LESSON 6"
6560 PRINT
6570 PRINT"This is the first part of a two part lesson"
6580 PRINT"It is divided into the following sections."
6590 PRINT
6600 PRINT"1)  Introduction          4)  Concatenation"
6610 PRINT"2)  String Assignment     5)  String Functions"
6620 PRINT"3)  String ARRAYs"
6630 PRINT
6640 PRINT
6650 RETURN
6660 GOSUB 6510
6670 PRINT"Which do you wish to do?"
6680 PRINT
6690 PRINT"A  Continue on"
6700 PRINT"B  Review this section again"
6710 PRINT
6720 INPUT"press the letter opposite the correct answer and press ENTER";T$
6730 IF T$ = "A" OR T$ = "B" THEN RETURN
6740 GOTO 6660
6750 REM **
6760 REM ** subroutine for string assignment example
6770 REM **
6780 PRINT"10  READ A$,B$"
6790 PRINT"20  PRINT A$"
6800 PRINT"30  A$ = B$"
6810 PRINT"40  PRINT A$"
6820 PRINT"50 DATA "CHR$(34)"THE GRINCH"CHR$(34)","CHR$(34)"IS COMING"CHR$(34)
6830 RETURN
6840 REM **
6850 REM ** subroutine for arrays example
6860 REM **
6870 PRINT"10 CLEAR 2000"
6880 PRINT"20 DIM A$(100)"
6890 PRINT"30 X = 0"
6900 PRINT"40 X = X+1"
6910 PRINT"50 INPUT "CHR$(34)"ENTER up to 99 strings, ENTER 'END' to stop"CHR$(34)";A$(X)"
6920 PRINT"60 IF A$(X) = "CHR$(34)"END"CHR$(34)" THEN END"
6930 PRINT"70 GOTO 40"
6940 RETURN
6950 REM **
6960 REM ** this is subroutine for STRING FUNCTIONS
```

```
6970 REM **
6980 PRINT"10 A$ = "CHR$(34)"My aching fingers"CHR$(34)
6990 RETURN
7000 REM **
7010 REM **
7020 RUN "MENU"
7030 PRINT
7040 PRINT
7050 PRINT"Going to the next part, please standby"
7060 RUN "LESSON6A"
7070 END
```

```
 755 GOSUB 21000
 760 GOSUB 30000
 890 PRINT"A  I'm taking this part in its entirety."
 900 PRINT"B  I wish to review selected areas (or take the test)."
 902 PRINT"C  I want to go to the first part."
 904 PRINT"D  I want to return to the Menu."
 910 PRINT
 915 INPUT"Press either capital A, B, C, or D and then press ENTER";T$
 920 IF T$ = "D" GOTO 48000
 922 IF T$ = "C" GOTO 49000
 930 IF T$ = "B" GOTO 1000
 940 IF T$ <>"A" GOTO 890
 950 GOSUB 2000
 955 GOSUB 3000
 960 GOSUB 4000
 975 GOSUB 6000
 976 GOSUB 7000
 978 GOSUB 8000
 979 GOSUB 10000
 980 GOSUB 21000
 990 GOTO 49000
1000 GOSUB 21000
1002 GOSUB 30000
1003 PRINT
1005 PRINT"Please type in the number beside the area you wish"
1010 PRINT"to review (1 through 7) and then press ENTER - press 0 and"
1015 PRINT"press ENTER to return to the Menu."
1025 PRINT
1030 INPUT"What is your choice";N
1040 IF N = 0 GOTO 48000
1050 ON N GOSUB 2000,3000,4000,6000,7000,8000,47000
1060 GOTO 1000
2000 GOSUB 21000
2010 PRINT"                         Introduction"
2020 PRINT
2030 PRINT"This part of the computer assisted instruction program has "
2040 PRINT"nothing to do with BASIC.  Instead, it is about the Microsoft"
2050 PRINT"Editor which allows you to edit BASIC program statements so you"
2060 PRINT"don't have to retype a whole BASIC line just because of one"
2070 PRINT"typo.  Using the editor makes it very easy to alter the line."
2080 PRINT
2090 PRINT"Until you get familiar with the editor, you may wish to make a"
2100 PRINT"little 'cheat sheet' so you can have the commands available for"
2110 PRINT"quick reference.  Throughout this part you should have your"
2120 PRINT"manual open to the editor portion so you can follow along."
2130 PRINT
```

```
2140 PRINT
2150 INPUT"press ENTER";T$
2160 GOSUB 21000
2170 PRINT"                              INTRODUCTION"
2180 PRINT
2190 PRINT"The object of this half will be to get you familiar with the"
2200 PRINT"editor's key commands.  The lesson will not teach you all the"
2210 PRINT"commands available.  However, the core of knowledge it gives"
2220 PRINT"you will let you start editing BASIC programs. For some of the"
2230 PRINT"more sophisticated commands, you should refer to your Micro-"
2240 PRINT"soft Manual."
2250 PRINT
2260 INPUT"press ENTER";T$
2270 RETURN
3000 GOSUB 21000
3010 PRINT"                              Starting"
3020 PRINT
3030 PRINT"The editor is line oriented , meaning that you operate"
3040 PRINT"on one line at a time (and not on a screen of data like you do"
3050 PRINT"with a word processor).  "
3060 PRINT
3070 PRINT"Throughout this lesson, we will be using one example line to"
3080 PRINT"illustrate all the commands.  That line is listed below."
3090 PRINT
3100 PRINT"10 FOR X = 6*TO7 :PRINT X: NEXT Z"
3110 PRINT
3120 PRINT"The colons form what is called a MULTI STATEMENT line. Each "
3130 PRINT"time a colon is entered, the computer treats the data following"
3140 PRINT"it as a new line.  Therefore, on the above line we have three"
3150 PRINT"statements.  Obviously, there are several errors in the line."
3160 INPUT"press ENTER";T$
3170 GOSUB 21000
3180 PRINT"                          Starting"
3190 PRINT
3200 PRINT"10 FOR X = 6*TO77 :PRINT X: NEXT Z"
3210 PRINT
3220 PRINT If we wanted to EDIT this line, we would type in the word EDIT,
3240 PRINT"followed by the line number.  In this case, we would type in"
3250 PRINT"EDIT 10.  You may enter the EDITor in other ways, but for this"
3260 PRINT"lesson we will always use the EDIT (line number) syntax."
3265 PRINT"Always access the editor from the IMMEDIATE mode."
3270 PRINT
3280 PRINT"If you want to EXIT the EDITor after you are done editing, then"
3290 PRINT"you just press ENTER.  Pressing ENTER from the EDITOR mode"
3300 PRINT"updates the line, and puts you back in IMMEDIATE mode again."
3310 PRINT
```

```
3330 INPUT"press ENTER";T$
3340 GOSUB 21000
3350 PRINT"                    Starting"
3360 PRINT
3370 PRINT"10 FOR X = 5*to77 :PRINT X: NEXT X"
3380 PRINT
3390 PRINT"EDIT 10"
3400 PRINT"10"
3410 PRINT
3420 PRINT"In the above example, we typed in the word EDIT 10, and the"
3430 PRINT"computer put us in the EDIT mode.  Note that the line number"
3440 PRINT"we are EDITing appeared.  The cursor would normally be flashing"
3450 PRINT"just to the right of the line number."
3460 PRINT
3470 INPUT"press ENTER";T$
3480 GOSUB 21000
3490 PRINT"Type in the command to edit line 10"
3500 PRINT
3510 INPUT"ENTER the command now";T$
3520 PRINT
3530 IF T$ <> "EDIT 10" THEN PRINT"WRONG - the correct answer is EDIT 10.....TRY AGAIN":PRINT:INPUT"press
     ENTER";T$:GOTO 3480
3540 PRINT"GOOD .... now"
3550 PRINT"you are in the EDIT mode, what key do you press to exit the"
3560 PRINT"EDIT mode? ... Press the key now ... DON'T PRESS 'BREAK or'"
3565 PRINT"'CONTROL' D'"
3570 PRINT
3580 PRINT"EDIT 10"
3590 PRINT
3600 PRINT"10";
3610 A$ = INKEY$
3620 IF A$ = "" THEN GOTO 3610
3630 IF ASC(A$) <> 13 THEN PRINT:PRINT"WRONG - you should have pressed ENTER":PRINT"TRY
     AGAIN":PRINT:INPUT"press ENTER to start over";T$:GOSUB 21000:GOTO 3550
3640 PRINT" FOR X = 5*to77 :PRINT X: NEXT X"
3650 PRINT
3660 PRINT"Note how the rest of the line appeared after you pressed ENTER"
3670 PRINT"That's the way it would happen if you were actually doing it."
3680 PRINT
3690 INPUT"press ENTER";T$
3700 RETURN
4000 GOSUB 21000
4010 PRINT"                 nSPACEBAR, nSearch"
4020 PRINT
4030 PRINT"To move the cursor over the line you are editing, you simply"
4040 PRINT"press the spacebar, and it will move over one character."
```

```
4050 PRINT
4060 PRINT"For example, let's say you entered the edit mode and the cursor"
4070 PRINT"is just on the right of the line number. Go ahead and press the"
4080 PRINT"spacebar until you get to the end of the line."
4085 PRINT"(after the line is printed, press spacebar once more to go on)"
4100 PRINT
4105 Q$ = "FOR I = 0 TO 77 :PRINT "; NEXT I"
4110 I = 1
4117 PRINT"10 "Q$
4120 PRINT"EDIT 10"
4130 PRINT"10 ";
4135 A$ = INKEY$: IF A$ = "" THEN GOTO 4135
4137 IF I > LEN(Q$) THEN GOTO 4200
4140 IF A$ = CHR$ (32) THEN PRINT MID$(Q$,I,1);:I = I+1:GOTO 4135
4150 GOTO 4135
4200 PRINT
4205 PRINT
4210 PRINT"GOOD .. Do you see how that worked?"
4215 PRINT"Normally, your cursor would also be flashing, and you would"
4216 PRINT"still be in EDIT mode, with your cursor on the last character."
4220 PRINT
4230 INPUT"press ENTER"Y$
4240 GOSUB 21000
4250 PRINT"                    nSPACEBAR, nSearch
4260 PRINT
4270 PRINT"Notice the lowercase 'n' in the title to this section.  That"
4280 PRINT"refers to a feature of the editor that allows you to position"
4290 PRINT"the cursor 'n' spaces to the right.  That way you won't have"
4300 PRINT"to bound away at your spacebar to get to the 200th character"
4310 PRINT"on a long line.  All you do is press the number of characters"
4320 PRINT"you want to 'spacebar' over and then press the spacebar."
4330 PRINT"In this example, you are in the edit mode and you want to go"
4340 PRINT"over a few spaces."
4350 PRINT
4355 PRINT"Press a number (up to 255) , and then press the spacebar"
4360 PRINT
4362 PRINT"10 "Q$
4365 PRINT EDIT 10"
4370 PRINT"10 ";
4375 N$=""
4380 A$ = INKEY$:IF A$ = "" THEN GOTO 4380
4385 IF A$ = CHR$(32) AND N$ = "" THEN N$ = "1": GOTO 4410
4387 IF A$ = CHR$(32) AND LEN(N$) = 1 THEN GOTO 4430
4390 IF  ASC(A$)-47> 10 OR (ASC(A$)-47)< THEN GOTO 4380
4400 IF N$ = "" THEN N$ = A$: GOTO 4380
4405 N$ = N$ + A$
```

```
4410 A$ = INKEY$: IF A$ = "" THEN GOTO 4410
4420 IF A$ <> CHR$(32) THEN GOTO 4410
4430 PRINT LEFT$(O$,VAL(N$))
4440 PRINT
4450 PRINT"WHEW! That was quick!.. If you want to do it again, ENTER a 'Y'"
4460 INPUT"else ENTER an 'N'";T$
4470 IF T$ = "Y" THEN GOSUB 21000:GOTO 4355
4480 IF T$ <> "N" THEN GOSUB 21000: GOTO 4450
4490 GOSUB 21000
4500 PRINT"                    nSPACEBAR, nS(earch)"
4510 PRINT
4550 PRINT"If you entered a bigger number that your line length, then"
4560 PRINT"the computer just defaulted to the maximum line length."
4570 PRINT
4575 PRINT"Also, if you were actually using the editor, you would stay"
4576 PRINT" in the EDIT mode, with the cursor over the nth character, after"
4577 PRINT"you pressed the spacebar."
4578 PRINT
4580 PRINT"Oh, we managed to move the spacebar around a little, what if"
4590 PRINT"we wanted to find a specific character in the line, and we"
4600 PRINT"weren't quite sure exactly how far down the line it was?"
4610 PRINT"In that case we would use the nS(earch) feature."
4620 PRINT
4630 INPUT"press ENTER";T$
4640 GOSUB 21000
4650 PRINT"                    nSPACEBAR, nS(earch)"
4670 PRINT"With the Search feature, right after you type in EDIT 'line'"
4680 PRINT"number', you can drive the cursor to any letter in the line"
4690 PRINT" and if the letter is not in the line, then the editor defaults"
4700 PRINT"to the end of the line.  The editor only searches to the"
4710 PRINT" right of the cursor."
4720 PRINT
4730 PRINT"For example, you just typed in EDIT 10 and you wish to find"
4740 PRINT"the letter I. All you do is press 'S' and then press I. The"
4750 PRINT"editor recognizes upper and lower case, so be sure the case is"
4760 PRINT"right!.  Go ahead and do it now.  You may search for any"
4770 PRINT"letter, just type S first, then type the letter "
4780 PRINT
4785 PRINT"10 "O$
4790 PRINT"EDIT 10"
4795 PRINT"10 ";
4810 A$ = INKEY$
4820 IF A$ = "s" THEN A$ = "S"
4830 IF A$ <> "S" THEN GOTO 4810
4840 A$ = INKEY$
4845 IF A$ = "" THEN GOTO 4840
```

```
4850 N = INSTR(O$,A$)
4860 IF N = 0 THEN N = 1
4870 PRINT LEFT$(O$,N-1)
4880 PRINT
4890 PRINT"WOW  Another quick one... ENTER 'Y' to do again, else ENTER an"
4900 INPUT"'N'";T$
4910 PRINT
4920 IF T$ = "Y" THEN PRINT"(Search for a ";:GOTO 4770
4930 IF T$ <> "N" THEN GOTO 4890
4940 GOSUB 21000
4950 PRINT"                    nSPACEBAR, nS(earch)"
4960 PRINT
4970 PRINT"You will have noticed that the cursor stopped BEFORE the "
4980 PRINT"character that you were searching for.  That is what is "
4990 PRINT supposed to happen.  In addition, if you asked to search for a"
4995 PRINT"non-existent character, the computer printed the whole line."
5000 PRINT
5010 PRINT"No doubt, you will also have noticed that there is a lower case"
5020 PRINT"'n' in the title nS(earch).  It simply means that you can"
5030 PRINT"search for the nth occurance of the specific character."
5040 PRINT"For example, if we were in the EDIT mode for line 10, and we"
5050 PRINT"wanted to find the second occurance of the letter 'R' then we"
5060 PRINT"would type '2SR' and the cursor would skip over to the R in"
5070 PRINT"PRINT 5."
5080 PRINT
5110 INPUT"press ENTER";T$
5120 GOSUB 21000
5130 PRINT"                    nSPACEBAR, nS(earch)"
5140 PRINT
5150 PRINT"Here is an example of the nS(earch) feature."
5160 PRINT
5165 PRINT"10 "O$
5170 PRINT"EDIT 10"
5180 PRINT"1          :'Now type 2SR to find 2nd occurance of R in 10"
5185 PRINT
5187 PRINT
5190 PRINT"10 FOR X = s*to77 :P"
5195 PRINT
5200 PRINT"The cursor would stop just before the nth occurance of R."
5210 PRINT
5220 INPUT"press ENTER";T$
5230 GOSUB 21000
5240 PRINT"What would you type to find the 2nd occurance of the character"
5250 PRINT"'T' in a line you were editing (assume you are already in the'
5260 PRINT" EDIT mode)"
5270 PRINT
```

```
5290 INPUT"ENTER your answer now";T$
5295 PRINT
5300 IF T$ = "2S7" THEN GOTO 5330
5310 PRINT"WRONG - you should type 2S7"
5320 GOTO 5340
5330 PRINT"CORRECT - GREAT!"
5340 PRINT
5350 INPUT"press ENTER";T$
5360 GOSUB 40000
5370 IF T$ = "B" THEN GOTO 4000
5380 RETURN
6000 GOSUB 21000
6005 D$ = "10 FOR X = s*to77 :PRINT Y : NEXT I"
6006 D1$ = "FOR X = 's'*"
6010 PRINT"                    nD(elete)"
6020 PRINT
6030 PRINT"Hopefully, you are catching on to the way the boys and girls at"
6040 PRINT"Microsoft are doing things, and won't have too much trouble"
6050 PRINT"with this command.  It does what it looks like it does.  It"
6060 PRINT"deletes characters, one at a time, or 'n' at a time."
6070 PRINT
6080 PRINT"For example, if you are in the EDIT mode for line 10 and you "
6090 PRINT"want to delete the NEXT character, then just press D and the"
6100 PRINT"character will be enclosed in exclamation marks.  The exclama-"
6110 PRINT"tion marks indicate that if you don't change things, then the"
6120 PRINT"new line will not have the character in it."
6130 PRINT
6140 INPUT"press ENTER";T$
6150 GOSUB 21000
6160 PRINT"An example of the D(elete) command would be:"
6170 PRINT
6200 PRINT"10 FOR X = s*to77 :PRINT Y :NEXT I"
6210 PRINT"EDIT 10"
6220 PRINT"10                  :'now say we want to delete the 's*' chars"
6230 PRINT"                    :'all we do is put the cursor to the left"
6240 PRINT"                    :'of the two characters and press D twice"
6250 PRINT"10 FOR X = 's'*     :'and it would look like this"
6260 PRINT"                    :'then we would press ENTER and we would"
6262 PRINT
6270 PRINT"10 FOR X = to77 :PRINT Y :NEXT I      :'have this'
6280 PRINT
6290 PRINT"Study this example and read the appropriate paragraph in your"
6300 PRINT"manual."
6310 PRINT
6320 INPUT"press ENTER";T$
6340 GOSUB 21000
```

```
6350 PRINT"                          nD(elete)"
6360 PRINT
6370 PRINT D$
6380 PRINT
6390 PRINT"Now it is your turn.  You have to EDIT the above line so that"
6400 PRINT"the characters 's#' are deleted.  You must delete them one at a"
6410 PRINT"time (as we showed you in the previous example).  First type"
6420 PRINT"the appropriate command for editing line 10, then move the '
6430 PRINT"spacebar over to the appropriate place, then delete the two"
6440 PRINT"offending characters, then press ENTER'  (use upper case)"
6450 PRINT
6460 INPUT"ENTER the first command now";T$
6470 PRINT
6480 IF T$ <> "EDIT 10" THEN PRINT"WRONG - you should type in EDIT 10 first":PRINT:GOTO 6460
6490 PRINT"10 ";
6499 I = 0
6500 A$ = INKEY$
6510 IF A$ = "" THEN GOTO 6500
6515 IF A$ = "d" THEN A$ = "D"
6520 IF A$ = CHR$(13) AND I <> 3 THEN I = I + 1:PRINT MID$(D1$,I,1);:GOTO 6500
6530 IF A$ = "D" AND I <> 3 THEN PRINT:PRINT"WRONG - don't press D until just before the
     's'":PRINT:PRINT"START OVER": PRINT:PRINT D$:GOTO 6450
6540 IF A$ = "D" AND I = 3 THEN PRINT MID$(D1$,4,2);:I = 12: GOTO 6500
6545 IF A$ = "D" AND I = 12 THEN PRINT MID$(D1$,12,2);:I = 15:GOTO 6500
6550 IF ASC(A$) = 13 AND I <> 15 THEN PRINT:PRINT"WRONG - do not hit ENTER until you are
     done":PRINT:PRINT"TRY AGAIN":PRINT:GOTO 6450
6560 IF ASC(A$) = 13 AND I = 15 THEN GOTO 6600
6570 GOTO 6500
6600 PRINT"to77 :PRINT x :NEXT I"
6605 PRINT
6608 PRINT"The line in the computer's memory would now look like this:":PRINT "10 FOR I = to77 :PRINT x
     :NEXT I"
6609 PRINT
6610 PRINT"GREAT  ... If you want to do it again, press 'Y' else press 'N'"
6620 INPUT"ENTER your choice now (Y or N)";T$
6630 IF T$ = "Y" THEN GOTO 6340
6640 GOSUB 21100
6650 PRINT"                    nD(elete)"
6660 PRINT
6670 PRINT"For the example, we protected you from mistakes by ignoring"
6680 PRINT"some commands, and telling you what you did wrong for others."
6690 PRINT"If you really are editing a line, be sure you press the right"
6700 PRINT"buttons, because you won't get warning messages' (however."
6710 PRINT"you seldom ruin what you have done, the editor is very for-"
6720 PRINT"giving, it usually leaves you something, even when you make a"
6730 PRINT"boo-boo'"
```

```
6740 PRINT
6750 PRINT"As you may have guessed, the 'n' symbol in the title indicates"
6760 PRINT"that you may delete 'n' characters at a time.  If you press a"
6770 PRINT"number and press D, then that is how many characters will be"
6780 PRINT"deleted."
6790 PRINT
6800 INPUT"press ENTER";T$
6810 GOSUB 21000
6820 PRINT 3$
6830 PRINT
6840 PRINT"If we wanted to delete 's#' all a once, we could position the"
6850 PRINT"cursor to just before the 's' and type in 2D and both "
6860 PRINT"characters would appear like this: !s#!"
6870 PRINT
6880 PRINT"Note the exclamation marks, in this case, are around both"
6890 PRINT"characters.  When you get through here, practice with some"
6900 PRINT"lines you have arbitrarily made up. You will see that this"
6910 PRINT"command can be very handy."
6920 PRINT
6930 INPUT"press ENTER";T$
6940 GOSUB 40000
6950 IF T$ = "B" THEN GOTO 6000
6960 RETURN
7000 GOSUB 21000
7010 PRINT"                    X(tend line)"
7020 PRINT
7030 PRINT"This is one of the easiest, and most useful of the commands"
7040 PRINT"It allows you to start up at the end of a line, just as if you"
7050 PRINT"never pressed ENTER."
7060 PRINT
7070 PRINT"First, you go to EDIT mode, then you press X. When you do, you"
7080 PRINT"will see the whole line displayed, and you can add anything"
7090 PRINT"on to the END of it.   Try it now.  First, type the command to"
7100 PRINT"get into EDIT mode for line 10, then press X, then type in"
7110 PRINT"anything you want (most micros allow a maximum of 249"
7120 PRINT"characters).  Then press ENTER.  Do it now."
7130 PRINT
7140 O$ = "FOR X = to77 :PRINT Y :NEXT I"
7150 INPUT"ENTER the first command";T$
7160 IF T$ <> "EDIT 10" THEN PRINT:PRINT  "WRONG - you should type in EDIT 10 first":PRINT:GOTO 7150
7165 PRINT
7170 PRINT"10 ";
7180 A$ = INKEY$
7185 IF A$ = "x" THEN A$ = "X"
7190 IF A$ <> "X" THEN GOTO 7180
7200 PRINT O$;
```

```
7210 LINE INPUT T$
7230 GOSUB 21000
7240 PRINT"Your new line, which was the sum of the old line 10 plus the"
7250 PRINT"data you typed in, is now this:"
7260 PRINT
7270 PRINT O$ + T$
7280 PRINT
7290 PRINT"Neat huh?  You will find yourself using this command the most."
7291 PRINT
7292 PRINT"As usual, we protected you from making mistakes, by only"
7293 PRINT"allowing you to execute the X(tend) command.  Remember, you"
7294 PRINT"will have much more freedom if you really are in the editor"
7295 PRINT"mode.  In fact, in the real editor, you can use the back arrow"
7296 PRINT"to wipe out the end of the line you are editing, and replace"
7297 PRINT"the old data with new stuff.  Be sure to practice this and"
7299 PRINT"you will grow to love it! -- ...well, maybe just like it a lot."
7300 PRINT
7310 INPUT"press ENTER";T$
7320 GOSUB 40000
7330 IF T$ = "B" THEN GOTO 7000
7340 RETURN
8000 GOSUB 21000
8010 PRINT"                    nC(hange) & I(nsert)"
8020 PRINT
8030 PRINT"Now for the meat of this half!  We are going to change the "
8040 PRINT"nasty errors in our trial statement, and then Insert some"
8050 PRINT"correct figures."
8060 PRINT
8070 O1$ = "FOR X = to77 :PRINT Y :NEXT Z"
8080 PRINT O1$
8090 PRINT
8100 PRINT"Above is our line (with the 's#' characters missing - we "
8110 PRINT"deleted them in the nD(elete) section).  Let's say we want"
8120 PRINT"to change 'NEXT Z' to 'NEXT X' and 'PRINT Y' to 'PRINT X'"
8130 PRINT"In addition, we want the value of X to start at 1 in the FOR"
8140 PRINT"NEXT loop (ie. we want to Insert a 1 just before the to77)"
8150 PRINT
8160 INPUT"ore:s ENTER to start our EDITing";T$
8170 GOSUB 21000
8180 PRINT"                    nC(hange) & I(nsert)"
8190 PRINT
8200 PRINT O1$
8210 PRINT
8220 PRINT"First we'll change 'PRINT Y' to 'PRINT X'"
8230 PRINT
8240 PRINT"To use the nC(hange) command, get into the EDIT mode and "
```

```
8250 PRINT"position the cursor to JUST BEFORE the character to be changed"
8260 PRINT"then press 'C' and press the new character, then press ENTER."
8270 PRINT"Try it now.  Get to EDITor, SPACE over to just before the Y"
8275 PRINT"and type in 'C', then type in an X (we want to swap Y with X)"
8280 PRINT"and finally, press ENTER  (use capitals)"
8290 PRINT
8300 PRINT"ENTER the first command at the bottom of the next line"
8302 PRINT"10 "Q1$
8304 LINE INPUT T$
8310 IF T$ <> "EDIT 10" THEN PRINT"WRONG - you have to type EDIT 10 first":PRINT: GOTO 8300
8320 PRINT
8325 I = 0
8330 PRINT"10 ";
8340 A$ = INKEY$
8350 IF A$ = "" THEN GOTO 8340
8355 IF A$ = "c" THEN A$ = "C"
8360 IF A$ = CHR$(32) AND I < 20 THEN I = I + 1: PRINT MID$(Q1$,I,1);:GOTO 8340
8370 IF A$ = "C" AND I <> 20 THEN PRINT:PRINT"WRONG - you must press 'C' just before the Y in PRINT
     Y":PRINT"TRY AGAIN":PRINT: INPUT"press ENTER";T$:PRINT:GOTO 8170
8380 IF A$ = "C" AND I = 20 THEN GOTO 8400
8390 GOTO 8340
8400 A$ = INKEY$
8410 IF A$ = "" THEN GOTO 8400
8415 IF A$ = "x" THEN A$ = "X"
8420 IF A$ = "X" THEN GOTO 8450
8430 GOTO 8400
8450 PRINT"X";
8460 A$ = INKEY$
8480 IF A$ <> CHR$(13) THEN GOTO 8460
8490 PRINT" :NEXT Z"
8500 PRINT
8510 PRINT"How's that for class?  Remember, we protected you from mistakes."
8520 PRINT"The real editor will do whatever you tell it, even if it is"
8530 PRINT"wrong'.   But you knew that, didn't you?"
8540 PRINT
8550 INPUT"ENTER a 'Y' if you want to do this again, else ENTER an 'N'";T$
8560 IF T$ = "Y" THEN GOTO 8170
8570 IF T$ <> "N" THEN GOTO 8550
8580 GOSUB 21000
8590 PRINT"                    nC(hange) & I(nsert)"
8600 PRINT
8610 PRINT"As with the other commands, the 'n' in nC(hange) designates"
8620 PRINT"how many characters are affected by the command.  If you want"
8630 PRINT"to change 10 characters, then you would type '10C' in the EDIT"
8640 PRINT"mode, and you would then HAVE to change the next 10 characters."
8670 PRINT
```

```
8680 PRINT"What would you type if you were in the EDIT mode and your "
8690 PRINT"cursor was just before a block of 4 characters that you wanted"
8700 PRINT"to change to 'XXXX'? "
8710 PRINT
8720 INPUT"ENTER your answer now";T$
8730 PRINT
8740 IF T$ <> "4CXXXX" THEN PRINT"WRONG - you should have typed 4CXXXX":PRINT:PRINT"TRY
     AGAIN":PRINT:INPUT"press ENTER";T$:GOTO 8670
8750 PRINT"GREAT! Now you have the idea!"
8760 PRINT
8770 INPUT"press ENTER";T$
8780 GOSUB 21000
8790 PRINT"                          nC(hange) & I(nsert)"
8800 PRINT
8810 PRINT"10 FOR X = to77 :PRINT X :NEXT Z"
8820 PRINT
8830 PRINT"We would change the 'Z' to an 'X' in the same way, but, to"
8840 PRINT"save time, we'll envoke some magic, and change it now so we can"
8850 PRINT"get to the I(nsert) command!    READY?   zzzzzZZZZAAAPPP!"
8860 PRINT"(<< POOF >>).  There, it's changed now.  Look below."
8870 PRINT
8880 PRINT"10 FOR X = to77 :PRINT X :NEXT X"
8890 PRINT
8900 PRINT"How would you like to have THAT editor at your command?"
8920 PRINT"We'll now get to the I(nsert) command.  Remember, we want to"
8930 PRINT"insert a 1 just before the 'to77'."
8960 PRINT
8970 INPUT"press ENTER for the Insert example";T$
8980 GOSUB 21000
8990 PRINT"                     nC(hange) and I(nsert)"
9000 PRINT
9010 Q$ = "FOR X = to77 :PRINT X :NEXT X"
9020 PRINT"10 "Q$
9030 PRINT
9060 PRINT"To use the I(nsert) command, you first get into the EDIT mode"
9070 PRINT"and then place the cursor to just before the character you want"
9080 PRINT"to insert the data in front of."
9090 PRINT
9100 PRINT"In this case, we get into the EDIT mode, then"
9110 PRINT"we SPACE over to just before the 'to77' and then we type an 'I'"
9120 PRINT"for I(nsert).  After the 'I' command we want to put in a 1."
9125 PRINT"but we COULD type in as many characters we want ...."
9130 PRINT"until we press ENTER.  At that time, all our changes are made"
9140 PRINT"and we are returned to the IMMEDIATE mode."
9150 PRINT
9160 INPUT"press ENTER to start the example";T$
```

```
9170 GOSUB 21000
9180 PRINT"Remember, first type EDIT 10, then space over to just before"
9190 PRINT"the 'to77', then type 'I', then type a 1, then type ENTER."
9200 PRINT
9210 PRINT"10 "0$
9220 PRINT
9230 INPUT"ENTER the first command";T$
9240 IF T$ <> "EDIT 10" THEN PRINT "WRONG - you must type EDIT 10 first":PRINT:GOTO 9230
9250 PRINT
9260 PRINT"10 ";
9265 I = 0
9270 A$ = INKEY$
9280 IF A$ = "" THEN GOTO 9270
9285 IF A$ = "i" THEN A$ = "I"
9290 IF A$ = CHR$(32) AND I < 8 THEN I = I + 1:PRINT MID$(0$,I,1);:GOTO 9270
9300 IF A$ = CHR$(13) AND I < 9 THEN PRINT:PRINT"WRONG - don't press ENTER until you are
     done":PRINT:INPUT"press ENTER to start over";T$:PRINT:GOTO 9170
9310 IF A$ = "I" AND I < 8 THEN PRINT:PRINT"WRONG - don't type I until just before the
     'to77'":PRINT:INPUT"press ENTER to start again";T$:PRINT:GOTO 9170
9320 IF A$ = "I" AND I >=8 THEN GOTO 9400
9330 GOTO 9270
9400 A$ = INKEY$
9410 IF A$ <> "I" THEN GOTO 9400
9420 PRINT A$;
9430 A$ = INKEY$
9440 IF A$ <> CHR$(13) THEN GOTO 9430
9450 PRINT"to77 :PRINT X :NEXT X"
9460 PRINT
9470 INPUT"ENTER a 'Y' to do this again, else ENTER an 'N'";T$
9480 IF T$ = "Y" THEN GOTO 9170
9490 IF T$ <> "N" THEN GOTO 9470
9500 GOSUB 21000
9510 PRINT"                    nC(hange) & I(nsert)"
9520 PRINT
9530 PRINT"With some of the same magic we envoked before, we will also"
9540 PRINT"put spaces in the proper places of the test line, (again."
9550 PRINT"you would normally use your nC or I command to fix up the line"
9560 PRINT"but I feel pretty magical today, so I want to do it."
9570 PRINT
9580 PRINT"zzzIIIAAAAAPPPP <<< POOF >>>.....CRAASSSHHHH .. tinkle"
9590 PRINT"10FORGROUNDXTWIXASLFUO)(()***sluurrrrrpo.....oooooo'"
9600 PRINT
9610 PRINT"Ok well, we'll leave it up to you to do in your practice"
9620 PRINT"sessions."
9630 PRINT
9640 INPUT"press ENTER";T$
```

```
 9650 GOSUB 40000
 9660 IF T$ = "B" THEN GOTO 8000
 9670 RETURN
10000 GOSUB 21000
10010 PRINT"You have finished the lesson and you can now take the test."
10100 PRINT"If you wish to review parts of the lesson, ENTER an 'R'"
10200 PRINT"else, if you want to continue to the test ENTER a 'C'"
10300 PRINT
10400 INPUT"ENTER your choice now (R or C)";T$
10500 IF T$ = "R" THEN RUN
10600 IF T$ <> "C" THEN GOTO 10000
10700 GOTO 47000
20980 REM **
20990 REM ** This subroutine clears the screen on any terminal
20995 REM **
21000 FOR X = 1 TO 24
21010    PRINT
21020 NEXT X
21030 RETURN
30000 PRINT"                    LESSON 6"
30010 PRINT
30015 PRINT"This is the second part of a two part lesson"
30020 PRINT"It is divided into the following sections."
30025 PRINT
30030 PRINT"1)  Introduction          4)  nD(elete)"
30035 PRINT"2)  Starting (EDIT/exit)  5)  X(tend line)"
30040 PRINT"3)  nSPACEBAR, nS(earch)  6)  nC(hange), I(nsert)"
30060 PRINT"                  7)  TEST"
30090 PRINT
30100 RETURN
40000 GOSUB 21000
40005 PRINT"Which do you wish to do?"
40010 PRINT
40020 PRINT"A  Continue on"
40030 PRINT"B  Review this section again"
40040 PRINT
40050 INPUT"press the letter opposite the correct answer and press ENTER";T$
40060 IF T$ = "A" OR T$ = "B" THEN RETURN
40070 GOTO 40000
47000 PRINT
47010 PRINT
47020 PRINT"Going to test.  Please wait one moment."
47030 RUN "TEST6"
48000 RUN "MENU"
49000 RUN"LESSON6"
50000 END
```

```
1000 REM **
1010 REM ** LESSON:  TEST6                VERSION: 1 AUG 83
1020 REM ** AUTHOR:  CAPT DAN CREAGAN
1030 REM **          AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM **            N$(X) = NAMES ARRAY, USED TO READ IN SEQ-
1070 REM **                    UENTIAL NAMES, AND TO WRITE OUT
1080 REM **                    UPDATE NAMES.
1090 REM **            S(X)  = SCORES ARRAY - USED TO READ AND
1100 REM **                    WRITE SCORES
1110 REM **            Q(X)  = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM **                    CORRECT ANSWERS.  IF AN ARRAY
1130 REM **                    ELEMENT EQUALS 1, THE ANSWER WAS
1140 REM **                    CORRECT
1150 REM **
1160 CLEAR 3000
1170 GOSUB 4110
1180 DIM N$(1000)
1190 DIM Q(10)
1200 DIM S(1000)
1210 PRINT"                    FINAL TEST (lesson 6)"
1220 PRINT
1230 PRINT"This test consists of 10 questions, you must get 70 percent"
1240 PRINT"of them correct to pass.  (that's 7 right out of the 10 ques-"
1250 PRINT"tions).  Use only capital letters in your answers, don't"
1260 PRINT"include extra spaces or letters."
1270 PRINT
1280 PRINT"                  GOOD LUCK!!"
1290 PRINT
1300 INPUT"press ENTER to continue";T$
1310 GOSUB 4110
1320 PRINT"What is wrong with the following statement?"
1330 PRINT
1340 GOSUB 4230
1350 PRINT
1360 PRINT"A  Multi Statement lines are not allowed"
1370 PRINT"B  The assignment of values between A$ and B$ are not valid."
1380 PRINT"C  The strings were not initialized"
1390 PRINT"D  Nothing"
1400 PRINT
1410 GOSUB 4130
1420 PRINT
1430 IF T$ = "D" THEN GOTO 1470
1440 PRINT"WRONG - the correct answer is D"
1450 PRINT"        See part 1, String Functions"
```

```
1460 GOTO 1490
1470 PRINT"CORRECT"
1480 Q(1) = 1
1490 PRINT
1500 INPUT"press ENTER";T$
1510 GOSUB 4110
1520 GOSUB 4230
1530 PRINT
1540 PRINT"What is the output of the above program? (you may use your "
1550 PRINT"BASIC manual to look up terms)"
1560 PRINT
1570 INPUT"ENTER the output now EXACTLY as it would appear";T$
1580 PRINT
1590 IF T$ = "H" THEN GOTO 1640
1600 PRINT"WRONG - the correct answer is H"
1610 PRINT"        If A$ = H(1) and B$ = A$, then the left character"
1620 PRINT"        of B$ is an H. See part 1."
1630 GOTO 1660
1640 PRINT"CORRECT"
1650 Q(2) = 1
1660 PRINT
1670 INPUT"press ENTER";T$
1680 GOSUB 4110
1690 PRINT"What is wrong with the following program."
1700 PRINT
1710 PRINT"10 FOR X = 1 to 20"
1720 PRINT"20   A$(X) = "CHR$(34)"0"CHR$(34)
1730 PRINT"30 NEXT X"
1740 PRINT
1750 PRINT"A  The array is not dimensioned properly"
1760 PRINT"B  You cannot address a single dimensioned array with a loop"
1770 PRINT"C  The "0" should not be enclosed in quotes"
1780 PRINT"D  Nothing"
1790 PRINT
1800 GOSUB 4130
1810 PRINT
1820 IF T$ = "A" THEN GOTO 1860
1830 PRINT"WRONG - the answer is A (it should be DIMensioned to 20)"
1840 PRINT"        See part 1, String Arrays."
1850 GOTO 1880
1860 PRINT"CORRECT"
1870 Q(3) = 1
1880 PRINT
1890 INPUT"press ENTER";T$
1900 GOSUB 4110
1910 PRINT"What is the output of the following program?"
```

```
1920 PRINT
1930 GOSUB 4230
1940 PRINT"20 B$ = A$ + B$"
1950 PRINT
1960 INPUT"ENTER your answer EXACTLY as it would appear";T$
1970 PRINT
1980 IF T$ = "Hi!H" THEN GOTO 2030
1990 PRINT"WRONG - the correct answer is Hi!H"
2000 PRINT"        If A$ = Hi! and B$ in line 10  equals H, then"
2010 PRINT"        A$ + B$ = Hi!H. See part 1, String Arrays."
2020 GOTO 2050
2030 PRINT"CORRECT"
2040 Q(4) = 1
2050 PRINT
2060 INPUT"press ENTER";T$
2070 GOSUB 4110
2080 PRINT"What is the output of the following program?"
2090 PRINT
2100 PRINT"10 A$ = "CHR$(34)"SOMEWHERE"CHR$(34)":A$ = MID$(A$,1,4):PRINT A$"
2110 PRINT
2120 INPUT"ENTER your answer EXACTLY as it would appear";T$
2130 PRINT
2140 IF T$ = "SOME" THEN GOTO 2180
2150 PRINT"WRONG - the correct answer is SOME"
2160 PRINT"        See your BASIC manual."
2170 GOTO 2200
2180 PRINT"CORRECT"
2190 Q(5) = 1
2200 PRINT
2210 INPUT"press ENTER";T$
2220 GOSUB 4110
2230 PRINT"What is the command you would enter to edit line number 50 of"
2240 PRINT"a program?"
2250 PRINT
2260 INPUT"ENTER your answer EXACTLY as it would appear";T$
2270 PRINT
2280 IF T$ = "edit 50" THEN GOTO 2330
2290 IF T$ = "EDIT 50" THEN GOTO 2330
2300 PRINT"WRONG - the correct answer is EDIT 50"
2310 PRINT"        See part 2, EDIT"
2320 GOTO 2350
2330 PRINT"CORRECT"
2340 Q(6) = 1
2350 PRINT
2360 INPUT"press ENTER";T$
2370 GOSUB 4110
```

```
2380 PRINT"Assume you are in the EDIT mode.  You wish to place the "
2390 PRINT"cursor over the second occurance of the letter R in your line."
2400 PRINT
2410 PRINT"What is the command you would use?"
2420 PRINT
2430 PRINT"A  2SR"
2440 PRINT"B  2DR"
2450 PRINT"C  2RR"
2460 PRINT"D  2CR"
2470 PRINT"E  2 spacebar R"
2480 PRINT
2490 GOSUB 4180
2500 PRINT
2510 IF T$ = "A" THEN GOTO 2550
2520 PRINT"WRONG - the correct answer is A"
2530 PRINT"        See part 2, nS(earch)"
2540 GOTO 2570
2550 PRINT"CORRECT"
2560 Q(7) = 1
2570 PRINT
2580 INPUT"press ENTER";T$
2590 GOSUB 4110
2600 PRINT"Assume your are in the EDIT mode"
2610 PRINT
2620 PRINT"What is the command you would use to insert text starting"
2630 PRINT"where your cursor is now."
2640 PRINT
2650 INPUT"ENTER the command now";T$
2660 PRINT
2670 IF T$ = "I" OR T$ = "i" THEN GOTO 2710
2680 PRINT"WRONG - the correct answer is I"
2690 PRINT"        See Part 2, I(nsert)."
2700 GOTO 2720
2710 PRINT"CORRECT"
2720 Q(8) = 1
2730 PRINT
2740 INPUT"press ENTER";T$
2750 GOSUB 4110
2760 PRINT"Assume you are in the EDIT mode"
2770 PRINT
2780 PRINT"What is the command you would use to drive the cursor to the"
2790 PRINT"end of the line you are currently editing. (the command is"
2800 PRINT"one letter long)."
2810 PRINT
2820 INPUT"ENTER the command now";T$
2830 PRINT
```

```
2840 IF T$ = "X" OR T$ = "x" THEN GOTO 2890
2850 PRINT"WRONG - the correct answer is X"
2860 PRINT"       See part 2, X(tend)"
2870 GOTO 2900
2880 PRINT"CORRECT"
2890 Q(9) = 1
2900 PRINT
2910 INPUT"press ENTER";T$
2920 GOSUB 4110
2930 PRINT"Assume you have just finished a course in computer assisted"
2940 PRINT"instruction in BASIC.  What should you do?"
2950 PRINT
2960 PRINT"A Quit trying, now that you know how"
2970 PRINT"B Practice, practice, practice . . . . and enjoy, enjoy, enjoy"
2980 PRINT"C Sell yourself as a national treasure"
2990 PRINT"D Write a nasty letter to the author of the program"
3000 PRINT
3010 Q(10) = 1
3020 GOSUB 4180
3030 PRINT
3040 IF T$ = "D" THEN PRINT"DON'T BLAME ME ... I'M ONLY FOLLOWING ORDERS!":PRINT:INPUT"press ENTER";T$:
     GOTO 3120
3050 IF T$ = "C" THEN PRINT"That won't help the National Debt very much!":PRINT:INPUT"press ENTER";T$:GOTO
     3120
3060 IF T$ = "B" THEN PRINT"Don't try to butter me up, I know you're into masochism!":PRINT:INPUT"press
     ENTER";T$:GOTO 3120
3070 IF T$ = "A" THEN PRINT"Obviously we have failed to communicate, I'm reporting you to":PRINT"the FBI
     for tax evasion and mail fraud.": PRINT"Try AGAIN":PRINT:INPUT"press ENTER";T$:GOTO 2920
3080 PRINT"Congratulations, you are one of the few who selected an answer"
3090 PRINT"that wasn't listed. Were you ever an extra for the MUPPET SHOW?"
3100 PRINT
3110 INPUT"press ENTER";T$
3120 GOSUB 4110
3130 PRINT"Obviously, you get automatic credit for the last question."
3140 PRINT
3150 PRINT"It was nice doing business with you. So long!"
3160 PRINT
3170 INPUT"press ENTER";T$
3180 GOSUB 4110
3190 FOR X = 1 TO 10
3200    Y = Y+Q(X)
3210 NEXT X
3220 PRINT"You have finished the test, out of 10 possible correct answers"
3230 PRINT"you scored "Y"."
3240 PRINT
3250 IF Y > 5 THEN  PRINT"YOU HAVE PASSED"
```

```
3260 GOSUB 3920
3270 IF Y > 6 THEN GOTO 3350
3280 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
3290 PRINT
3300 PRINT"YOU SHOULD RETAKE LESSON 6!"
3310 PRINT
3320 PRINT"You will be returned to the Menu."
3330 PRINT
3340 GOTO 4250
3350 PRINT
3360 PRINT"Do you want your score recorded on a permanent file?"
3370 PRINT
3380 PRINT"A  YES"
3390 PRINT"B  NO"
3400 PRINT
3410 INPUT"Which";T$
3420 IF T$ = "B" THEN GOTO 3720
3430 GOSUB 4110
3440 PRINT"To record your score, we must open a file and put your name"
3450 PRINT"in it.  Therefore, surprisingly, we need your name.  If your"
3460 PRINT"name is not unique among the students likely to take this test,"
3470 PRINT"please contact your test monitor for an identifying word that"
3480 PRINT"will make you unique.  Then enter that word below."
3490 PRINT
3500 PRINT"If you have already entered a score previously, be sure to"
3510 PRINT"enter the same name you used before.  Use all capitals."
3520 PRINT
3530 INPUT"ENTER your word or name now";T$
3540 OPEN"I",1,"SCORES"
3550 X = 0
3560 IF EOF(1) THEN GOTO 3620
3570 X = X+1
3580 INPUT#1,N$(X)
3590 INPUT#1,S(X)
3600 IF N$(X) = T$ THEN GOTO 3300
3610 GOTO 3560
3620 CLOSE
3630 X = X+1
3640 N$(X) = T$
3650 S(X) = Y
3660 OPEN"O",1,"SCORES"
3670 FOR W = 1 TO X
3680   PRINT#1,N$(W)
3690   PRINT#1,S(W)
3700 NEXT W
3710 PRINT
```

```
3720 GOSUB 4110
3730 PRINT"You are FINISHED WITH THIS COURSE !!!   "
3740 PRINT
3750 PRINT"It was GREAT having you as a student - THANKS!"
3760 PRINT"When you press ENTER you will be sent to the Menu from "
3770 PRINT"where you can review other lessons or quit"
3775 PRINT
3780 PRINT"For homework, you may wish to change your inventory program"
3787 PRINT"so that it will handle string data.  That way you can include"
3790 PRINT"the names of your furniture in your file.  However, we leave"
3795 PRINT"that up to you. Hasta Luego!"
3798 PRINT
3799 GOTO 4250
3810 S(X) = Y
3815 IF EOF(1) THEN CLOSE:GOTO 3850
3820 X = X+1
3830 INPUT#1, N$(X), S(X)
3840 GOTO 3810
3850 OPEN"O",1,"SCORES"
3860    FOR W = 1 TO X
3870        PRINT#1,N$(W)
3880        PRINT#1,S(W)
3890    NEXT W
3900 PRINT
3910 GOTO 3730
3920 IF Y=10 THEN RETURN
3930 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
3940 PRINT
3950 IF Q(1)  = 0 THEN PRINT"  part 1. STRING ASSIGNMENT, STRING FUNCTIONS"
3960 IF Q(2)  = 0 THEN PRINT"  part 1. STRING ASSIGNMENT, STRING FUNCTIONS"
3970 IF Q(3)  = 0 THEN PRINT"  part 1. STRING ARRAYS"
3980 IF Q(4)  = 0 THEN PRINT"  part 1. CONCATENATION"
3990 IF Q(5)  = 0 THEN PRINT"  part 1. STRING FUNCTIONS"
4000 IF Q(6)  = 0 THEN PRINT"  part 2. EDIT"
4010 IF Q(7)  = 0 THEN PRINT"  part 2. nS(earch)"
4020 IF Q(8)  = 0 THEN PRINT"  part 2. I(nsert)"
4030 IF Q(9)  = 0 THEN PRINT"  part 2. F(ile)"
4040 PRINT
4050 INPUT"press ENTER";T$
4060 GOSUB 4110
4070 RETURN
4080 REM **
4090 REM ** this subroutine clears the screen
4100 REM **
4110 FOR I = 1 TO 24
4120     PRINT
```

```
***** Listing of Program 'TEST6' *****                    07/13/80 - 14:45:14

 4130 NEXT x
 4140 RETURN
 4150 REM **
 4160 REM ** this subroutine is for the response section
 4170 REM
 4180 INPUT"ENTER the letter opposite the correct answer"it$
 4190 RETURN
 4200 REM **
 4210 REM ** subr for string assignment/functions
 4220 REM **
 4230 PRINT"it A$ = "CHR$(34)"Hi"CHR$(34) ":B$ = A$ :S$ = LEFT$(B$,1. :PRINT B$"
 4240 RETURN
 4250 PRINT
 4260 INPUT"press ENTER to return to MENU"it$
 4270 RUN"MENU"
 4280 CLOSE
 4290 OPEN"I",1,"SCORES"
 4300  IF EOF(1) THEN STOP
 4310 INPUT#1,A$,N
 4320 PRINTA$,N
 4330 GOTO 4310
```

APPENDIX C

OUTPUT OF THE CAI PROGRAM LIBRARY

TABLE OF CONTENTS

```
TRSDOS Ready
BASIC
BASIC 01.00.00 for TRSDOS Version 6
Copyright (c) 1983 By Microsoft. Licensed to Tandy Corporation.
All rights reserved.

Ready
RUN"MENU"
```

COMPUTER ASSISTED INSTRUCTION IN BASIC

by: Captain Dan Creagan
Air Force Institute of Technology

This is the menu for computer assisted instruction in BASIC. It is meant to be used with a BASIC manual or with an experienced programmer available for consultation.

If you wish to give me feedback, or get information about this program, please contact me at GRIFFISS AFB, NY. I will be in the Aircraft Maintenance area after OCT 83.

press ENTER to continue?

MENU CHOICES

Select the program you wish to run from the list below and press the number that is next to your selection. Then press ENTER.

1. LESSON ONE	7. LESSON FOUR
2. TEST ONE	8. TEST FOUR
3. LESSON TWO	9. LESSON FIVE
4. TEST TWO	10. TEST FIVE
5. LESSON THREE	11. LESSON SIX
6. TEST THREE	12. TEST SIX

WHICH NUMBER DO YOU WANT? 1

```
LESSON:  BASIC 1A            VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON:  About One Hour


AUTHOR:  Capt Danny J. Creagan
         Air Force Institute of Technology

OBJECTIVE:  To introduce the student to Microsoft
            BASIC and the fundamentals of a small computer

MATERIALS REQ'D:  BASIC reference manual


press the ENTER key to continue?



                    LESSON 1

This is the first part of a two part lesson
It is divided into the following sections.

1)  Introduction          5)  Statements & Programs
2)  Hardware              6)  Print Statement
3)  Software              7)  End & Stop Statement
4)  General Information   8)  Immediate Mode, NEW
                              LIST, DELETE

A  I'm taking this part in its entirety.
B  I wish to review selected areas.
C  I want to go to the second part.
D  I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A



                    Introduction
```

Throughout all your lessons, you should have your BASIC manual handy. If you find yourself stumped by a question, you should LOOK UP THE ANSWER IN THE BOOK. If you can't find it after an honest attempt, then make a guess and then go on. You will have an opportunity to review each section again.

Beginning with this lesson, you will have homework assigned at the end of each test. If you do the homework, you will learn more, and, with the techniques you learn, you will find that you can tackle small programming jobs as soon as you complete the course.

press ENTER?

Introduction

Throughout the next six lessons you will be learning about computers and what they do. Although the course is titled 'Computer Assisted Instruction in BASIC', you will also need to learn the terminology of computers, not just the BASIC programming language. This first lesson will start with some fundamental ideas, and expand them as we go along.

We use computers to process DATA and give us answers to our problems. To process this DATA, we must communicate with the computer using two basic computer components. Those components are called: HARDWARE AND SOFTWARE.

press ENTER?

Which do you wish to do

A Continue on
B Review this section again

press the letter opposite your choice and press ENTER? A

HARDWARE

Hardware is the term used to describe the electrical and mechanical aspects of a computer. Hardware includes the parts you can physically touch on, or in, your computer.

One major piece of hardware is the central processing unit (CPU). The CPU is the computer's central electronic brain.

It performs all of the data operations and contains a storage area called MEMORY which is used for short term data retention during operations.

press ENTER to continue?

HARDWARE (cont)

PERIPHERAL DEVICES are additional units of equipment that support the computer. PERIPHERAL DEVICES are used for long-term or permanent storage, and they also let you communicate with the computer.

The computer 'talks' to you by using peripheral hardware units called OUTPUT devices. These can be TERMINALS, or LINE PRINTERS, or TAPES, or DISKS.

You 'talk' to the computer through units called INPUT devices.

press ENTER to continue?

HARDWARE (cont)

INPUT DEVICES may also be terminals, or tape, or disks, or, in some special cases, printers that have keyboards that are used as terminals.

INPUT and or OUTPUT DEVICES provide a physical communication

link between you and the computer.

Whenever you communicate, there must be something that transforms your physical request (key strokes) into electrical DATA that the CPU understands. Most of that job is done by SOFTWARE

press ENTER to continue^

Here is a little quiz - answer in capital letters and do not include extra spaces or words

What component (HARDWARE or SOFTWARE) is mainly used to transform your inputs into a form the CPU can understand^

WRONG - the correct answer is SOFTWARE

press ENTER to continue^

What do the letters CPU stand for^ CENTRAL PROCESSING UNIT

CORRECT - now we are learning something!

press ENTER to continue^

Which of the following can be considered an OUTPUT device^

A Terminals
B Tapes
C Disks
D ALL of the above

which letter do you select^ D

CORRECT

press ENTER to continue?

Which do you wish to do

A Continue on
B Review this section again

press the letter opposite your choice and press ENTER? A

SOFTWARE

Software is a collection of written rules that control the computer. Software can be divided into two types: USER PROGRAMS and OPERATING SYSTEMs.

A USER PROGRAM is the instructions that you write to the machine that tell it where your data is, what to do with it, and when to do it.

The OPERATING SYSTEM is the software that is the consciousness of the computer.

press ENTER to continue?

SOFTWARE (cont)

The OPERATING SYSTEM supervises the various capabilities of the computer and cannot be altered by the user. It OVERSEES the operation, and senses when a keystroke is made, a button is pushed, or a request made.

One part of the operating system is called the LANGUAGE PROCESSOR. The LANGUAGE PROCESSOR translates the instructions of a user-written program into electronic instructions that the computer can understand.

The rules, or grammar, that you use to write your software

are described by the kind of computer language you use.

press ENTER to continue?

SOFTWARE (cont)

Because user-written programs and operating systems are both designed by humans, it is possible to develop a human oriented language that both can use. That is, a language that lets you write programs using easily mastered rules and conventions that are also understood by the operating system. Once we get the operating system to understand the instruction, it can make the computer do its job.

press ENTER to continue?

It's time for another quiz!

Remember, use only capital letters and don't add unnecessary spaces or words.

press ENTER to continue?

Is a peripheral, such as a line printer, hardware or software?

A Hardware
B Software

Choose A or B - press the letter and then press ENTER

Which letter? A

CORRECT

press ENTER to continue?

An operating system translates user-written code into a form that the operator can understand. (TRUE or FALSE?)

A True
B False

Choose the letter corresponding to the correct answer

which letter (A or B)? B

You are RIGHT

press ENTER to continue?

Which do you wish to do

A Continue on
B Review this section again

press the letter opposite your choice and press ENTER? A

General Information

There have been many programming languages developed over the years. Many were designed to solve specific problems and they required a good deal of previous knowledge about computers.

BASIC, which stands for Beginner's All-purpose Symbolic Instruction Code, is a language that requires only a moderate understanding of how a computer works.

BASIC was developed at Dartmouth College for use by students

who were unfamiliar with computers and needed a language related to everyday speech.

press ENTER to continue?

General Information (cont)

BASIC is easier to master than most other languages, because its instructions are very similar to English grammar.

However, BASIC is not English. A computer must be instructed in precise terms, with no ambiguity. English has many synonymous and imprecise terms.

press ENTER for more?

General Information (cont)

To further explain the difference between BASIC and English, if you describe how to average numbers in English you might do it this way. (assuming the numbers below)

Add 19, 80, 50 100, and 56. Divide by 5.
Write the quotient as the answer.

A computer programmed in BASIC couldn't understand these instructions; however, the instructions that BASIC would use are very similar to these. BASIC just distills down the commands and eliminates all the ambiguity.
This average can be stated in one instruction called PRINT.

press ENTER for an example of the PRINT instruction?

General Information (cont)

The PRINT statement works like this. to find the average
of five numbers and write the result on your terminal.
you can use the following BASIC statement:

```
PRINT (19+80+50+100+66)/5
```

In this example, the BASIC verb PRINT tells the operating
system to write the instruction following it to the terminal.
The data, or recipients of the verb PRINT, are the numbers
and symbols to the right of the PRINT word. The symbols
are used the same way that you use them on a calculator.

press ENTER to continue with General Information?

BASIC is used by nearly every micro and mainframe computer.

There are many 'dialects' of BASIC and they are not all
compatible with each other. For instance, a BASIC program
written in Honeywell BASIC or Applesoft BASIC will not
run on your computer unless it is modified.

The name for the BASIC on your computer is Microsoft BASIC.
Microsoft BASIC is supported by more microcomputers than
any other dialect. It is extremely powerful, and matches
the computing capability of most other languages. It
is slower than some, but the slowness is relative (most
calculations only take milliseconds).

press ENTER?

Why is BASIC a good general programming language to learn?

A It can be used by most students and programmers
B You don't need to know a lot about computers to use it
C It is available on most computers
D ALL of the above

Press the letter corresponding to the correct answer
Be sure to enter only capital letters

What is your selection? D

CORRECT

press ENTER to continue?

Which do you wish to do

A Continue on
B Review this section again

press the letter opposite your choice and press ENTER? A

STATEMENTS and PROGRAMS

The instruction that we saw in the previous example is a one line command to the computer. When we combine several statements, we get a more useful COMPUTER PROGRAM.

The COMPUTER PROGRAM acts as a series of directions for the machine to follow.

The statements that make up the program are expressed as BASIC verbs which denote an action to be taken. THEY APPEAR SEQUENTIALLY ON NUMBERED PROGRAM LINES, usually along with the data that is to be acted upon.

press ENTER to continue with STATEMENTS & PROGRAMS?

Each BASIC statement consists of a specific arrangement of elements. These elements are shown below, in the order they MUST appear in an actual program line

STATEMENT (or LINE) NUMBER
- indicates the processing sequence of the statements
- always in ascending order.

BASIC WORD
- specifies the computer operation to be performed

PARAMETERS
- variables, or expressions, used to direct the operation performed by the statement.

press ENTER to continue?

STATEMENTS and PROGRAMS (cont)

Every statement must have a line number and these numbers range from 0 to 65529 in most microcomputers that support Microsoft BASIC. (Microsoft is the company that owns the copyright to the particular dialect of BASIC that runs on this machine). It is advisable to write program lines in increments of 10 to allow you to insert additional lines without having to renumber every statement line. The statements are executed in ascending numerical order, not in the order they were entered.

(look up the RENUM, or NAME command in your manual for more information)

press ENTER to continue?

STATEMENTS and PROGRAMS (cont)

The last statement of the program should be the END statement. This indicates that the program is complete. IT IS NOT NECESSARY, but it is a good practice to always put it in.

To get the program to execute you use the RUN command.

Now for a few questions to see how you are doing.

press ENTER for the questions?

```
A computer program is a series of ---------

A  Verbs
B  Words
C  Statements
D  Synonyms


Type in the letter opposite the correct answer and press ENTER? C

CORRECT

press ENTER to continue?



Which of the following is incorrect?

A  A BASIC word is a word that a BASIC processor understands
B  A statement can have no more than two line numbers
C  Data are the recipients of the action of BASIC verbs
D  Line Numbers are written sequentially.

Press the letter that is beside the correct answer
and then press ENTER

What is your choice? B

CORRECT - way to go!

press ENTER for the next question?



Is 350000 a valid statement number in Microsoft BASIC?

N  No
Y  Yes

press the letter beside the correct answer and then press ENTER? Y
```

WRONG - 350000 is too big. Remember, you can only go to 65529

press ENTER?

Which do you wish to do

A Continue on
B Review this section again

press the letter opposite your choice and press ENTER? A

PRINT STATEMENT

The BASIC word PRINT is a command that tells the computer
to output the data that follows to the computer terminal

This data can be numbers, variables, or strings.
(strings are combinations of words or numbers that are
to be printed without having any calculations done to them)

press ENTER?

You can control the output caused by the print statement
in two ways. If you just want what you ENTER printed
without any calculations done to it, then you enclose the
data after the PRINT command in quotation marks.

For Example:

```
10 PRINT "Go For It!"
20 END
RUN
```

In BASIC, if you type this in as shown, you get this result:

Go For It!

press ENTER?

PRINT (cont)

Another example would be:

```
10 PRINT "This is easy"
20 END
RUN
```

Which would result in:

This is easy

Notice that nothing is changed by the computer, the words that were commanded to be output were printed exactly as shown.

press ENTER for the second example?

PRINT (cont)

The second way the print statement is used to control output is by NOT enclosing the data in quotation marks. The data is then read by the computer and the computer tries to evaluate what the data means in mathematical terms. If you have entered data that cannot be mathematically manipulated, then you get an ERROR message.

press ENTER?

Here is an example of data in quotes, and data not in quotes
and what the output would look like:

```
10 PRINT "1+1"
20 PRINT 1+1
RUN

Results in:

1+1
2

press ENTER?
```

PRINT (cont)

```
10 PRINT "1+1"
20 PRINT 1+1

1+1
2
```

Note that the statement that had quotes was reprinted exactly
as it was typed in, without the quotes, while the second
statement was computed mathematically and a result was given.
The part of the first statement within quotes is called
a STRING - (remember?)

press ENTER?

PRINT (cont)

The format of the output of PRINT can be controlled using
commas. For Example:

```
10 PRINT "A",6+2,"B",7+1
20 END
RUN
```

Results in:

```
A             8             8             8


press ENTER?


A             8             8             8
```

Notice how the commas have caused the terms of the statement to be spaced across the screen. The spaces are similar to TABS on a typewriter, however, the comma reacts differently on different terminals. See your BASIC manual or ask your system operator how they react on your machine.

(commas usually cause 8 spaces between terms)

press ENTER?

PRINT (cont)

The PRINT statement also allows you to output blank lines. You print blank lines by typing in the line number and then a PRINT statement without an argument. For example:

```
10 PRINT "Now is the time to skip"
20 PRINT
30 PRINT "a line."
RUN
```

press ENTER for results?

```
Now is the time to skip
```

a line

press ENTER?

12+10

Which of the following statements would cause the above output?

A 10 PRINT "12+10"

B 20 PRINT 12+10

C 15 PRINT "TWELVE + TEN"

D 25 PRINT 12+10

press the letter that is beside the correct choice and ENTER? A

CORRECT - that was a key concept, congratulations!

press ENTER?

Write the statement that would cause a blank line to be printed
Use 10 for the line number and leave one blank space between
terms.

What is your answer? 10 PRINT

CORRECT

press ENTER to continue?

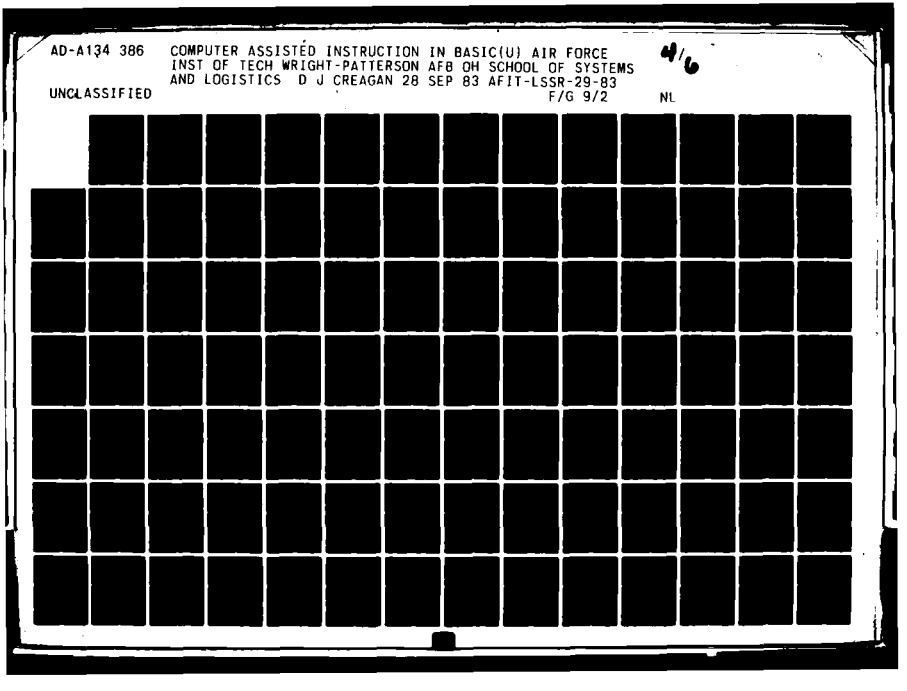
Which do you wish to do

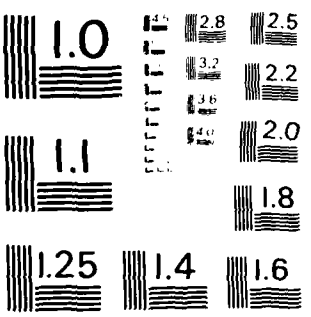
A Continue on
B Review this section again

AD-A134 386 COMPUTER ASSISTED INSTRUCTION IN BASIC(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS D J CREAGAN 28 SEP 83 AFIT-LSSR-29-83

4/6

UNCLASSIFIED F/G 9/2 NL





press the letter opposite your choice and press ENTER? A

END & STOP

The END statement is the last statement in a program.
It notifies the computer when the program is done.
Because it is the last statement, it has the highest
line number. The END statement is not necessary in Microsoft
BASIC, but many programmers use it anyway. They believe a
program is more understandable and easier to 'track' by
another programmer if there is only ONE entry and ONE exit in
a program.

press ENTER for the rest of END & STOP?

The STOP statement interrupts execution of the program.
It is primarily used as a debugging aid. If you want
to find the status of a variable at a certain point in a
program, you insert a STOP statement. For example:

```
10 X = 2+3
20 Y = X/5
30 STOP
40 X = Y+2
```

When this program is RUN it will STOP execution at line 30
Then you may ask the computer to tell you the status of any
of the variables X or Y. You can do this using the IMMEDIATE
mode (explained next section). Simply type in PRINT X,Y.

press ENTER?

If you want to start the program back up from where you
STOPed it, then type in CONT (CONTinue) and press ENTER
For example:

```
10 X = 12+3
20 Y = 3+2
30 STOP
40 PRINT X+Y
RUN

Results in:

BREAK IN 30


hit ENTER for the rest?



BREAK in 30


Now, by typing in CONT, the program will resume execution
For example:

CONT
22


Notice how the last line (which was PRINT (+Y) was executed?
It was just as if the STOP statement had never been there
As you progress in BASIC, you will find many uses for this
statement.

press ENTER to continue?



Which do you wish to do

A  Continue on
B  Review this section again

press the letter opposite your choice and press ENTER? A
```

IMMEDIATE MODE

Microsoft BASIC has a mode called IMMEDIATE. . . .
Whenever you have implemented BASIC in your system, usually by typing in the word BASIC, (consult your BASIC manual or your system operator for specific directions on your particular machine) you will be in the IMMEDIATE mode. In this mode, you may execute many BASIC statements without having to type in line numbers or the command RUN. For example:

PRINT "JUST WHEN I THOUGHT I HAD THE HANG OF IT"

This line will print the statement within the quotes as soon as the ENTER key is pressed.

press ENTER to continue?

IMMEDIATE (cont)

Another example would be:

PRINT 93+10+40

Which would result in:

143

As you can see, the computer will do the calculations just as if it were commanded to do it in the normal way.

press ENTER?

IMMEDIATE (cont)

The biggest disadvantage of the IMMEDIATE mode is that the data is not stored in memory, and cannot be repeated again. It is lost after the initial display, whereas the programs we looked at before can be run over and over again by merely

typing in the word RUN. Also, the immediate mode is limited to one line of statements at a time.

press ENTER?

NEW Statement

If you want to clean out the temporary memory in BASIC, all you have to do is type in the word NEW. BUT BE CAREFUL WITH THIS COMMAND. It will erase any program you have resident.

Let's say you've been practicing the commands you have learned so far, and you have put in a lot of line numbers and RUN them. But now you want to start over. You can erase the mess with the command NEW.

press ENTER?

LIST

Suppose you don't know what's there and you want to find out? Just type in the command LIST. LIST will show you everything That's in temporary memory.

press ENTER?

DELETE

Finally, what if you don't want to type in a new program, you just want to delete a line? You can do that by typing in DELETE 10, or DELETE 20 or DELETE (line number).

If you want to delete a range of line numbers, you type in DELETE (low range-high range). Say you want to delete lines 15 to 35. You would type in DELETE 15-35, and the lines

would no longer be in memory!

press ENTER?

Which of the following statements would be an example of
IMMEDIATE mode in Microsoft BASIC?

A 10 PRINT "AT LAST"
  RUN

B PRINT 25+2-3

C 32+2

D STOP

press the letter beside the correct answer and press ENTER? B

RIGHT - you've got the right idea about immediate mode

press ENTER to continue?

What command will erase everything in temporary memory?

Type in the command using capital letters? NEW

CORRECTO MUNDO - THAT'S RIGHT!

press ENTER?

What command will list all the line numbers and statements
that you have placed in temporary memory?

Type in the command using capital letters? LIST

RIGHT YOU ARE'

press ENTER?

What is the command to delete lines 20 to 50

A DELETE 20-50
B NEW
C DELETE 20 to 50
D ERASE 20 to 50

ENTER the correct answer? A

RIGHT AGAIN!

press ENTER?

Which do you wish to do

A Continue on
B Review this section again

press the letter opposite your choice and press ENTER? A

GOING TO SECOND PART - PLEASE STANDBY

LESSON 1B

This is the second part of a two part lesson
It is divided into the following sections.

1) Library Functions
2) Variables (general)
3) Numeric Variables
4) String Variables
5) Using Arithmetic
6) Comparing Variables & Lesson Summary
7) TEST

A I'm taking this part in its entirety.
B I wish to review selected areas (or take the test).

C I want to go to the first part.
D I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A

LIBRARY FUNCTIONS

Many mathematical FUNCTIONS such as square root, trignometric functions, and logrithms are difficult to derive using just addition, subtraction, multiplication, and division. To help us use these FUNCTIONS without deriving them from scratch each time we want to get a tangent or sine or square root, etc. Microsoft BASIC has a library of commonly used FUNCTIONS already programmed into permanent memory. All you have to do is call them with a BASIC command whenever you want to use them.

You identify which function you want to use by using a keyword, such as SQR for square root.

press ENTER?

LIBRARY FUNCTIONS (cont)

If you wanted to find the square root of 25, in the IMMEDIATE mode, you would type in:

PRINT SQR(25)

Which would result in:

5

Notice how the keyword precedes the value to be manipulated, and the value is enclosed in parenthesis?

press ENTER?

LIBRARY FUNCTIONS (cont)

Another example would be:

```
10 PRINT SQR(2+62)
RUN
```
Which would give you:

8

In this example, note that we applied a function to an expression with more than one term. This is entirely legal, and can shorten the number of statements you may need in your program.

press ENTER?

LIBRARY FUNCTIONS (cont)

You may use a function statement any number of times in your program. The different types of LIBRARY FUNCTIONS will be reviewed in a later lesson.

If you don't find the function you want in the library, then you may create your own function. This is called a USER DEFINED function. A USER DEFINED FUNCTION is not stored permanently in memory, it can only be used in the program it was created in. We will discuss USER DEFINED FUNCTIONS in a later lesson.

press ENTER?

QUIZ time! - USE ONLY CAPITAL LETTERS IN YOUR ANSWERS!

Are the library functions stored permanently in memory?

A YES

B No

press the letter opposite the correct answer and press ENTER? A

CORRECT

press ENTER to continue?

If the function for converting a number to an integer is INT, show the statement for finding the integer value of 27.56. Use line number 10, and leave only one space between elements. Do not include the RUN command.

DON'T FORGET TO ENCLOSE '27.56' IN PARENTHESIS

Type in your answer?

WRONG - the correct answer is --- 10 PRINT INT(27.56)

press ENTER to continue?

Which do you want to do?

A Continue on
B Review this lesson again

press the letter opposite your choice and press ENTER? A

Variables

When working with computers, it is necessary to define the type of data you are manipulating, if for no other reason than to communicate your program to someone else. Numbers, such as 10, 32, 50, 1, etc., are considered CONSTANTS. Can you guess why? Its because they never change, they are always worth a set amount. They are CONSTANT.

On the other hand, in algebra we learned that we could manipulate numbers and define problems easier if we assigned letters such as X and Y to equations. In this case, X and Y are VARIABLES. That is, they could assume any value we wanted as long as the value suited the equation.

press ENTER?

Variables (cont)

The way we treated letters in algebra, that is, assigning them values that were variable and were for calculation purposes, is the same way we treat them in the computer world.

For instance, if we give X the value of 8, then the computer will store the value 8 in a memory location that is labeled X. The value will not change until we assign a new value to the label X, or quit BASIC.

There are two fundamental types of variables in BASIC, they are NUMERIC variables, and STRING variables. Our previous example of assigning X a number made it a NUMERIC variable.

press ENTER?

If we had assigned a CHARACTER (such as my name, DAN) to a variable, then we would have created a STRING variable.

A STRING variable holds data that will not be operated on mathematically. (I wouldn't want my name operated on, would you?)

The reason for having STRING variables is so we can do things like print labels, make word processors, and develop computer assisted instruction programs. If these tasks are done in BASIC, then they are done using STRINGS.

press ENTER?

Which do you want to do?

A Continue on
B Review this lesson again

press the letter opposite your choice and press ENTER? A

Numeric Variables

In computers we assign values to variables to ease our job!

X = 1+2

In this case, the value of 3 would be assigned to X and the computer would store the value in its memory until we either changed it, or quit BASIC. In other words, we assigned the value of 3 to X, but only temporarily. Take the following example:

10 Y = 1+2
20 X = 4

What do you think the value of X is if we RUN the example?

press ENTER for the answer?

Numeric Variables (cont)

Of course, you knew the answer was 4, didn't you?

Because long programs sometimes need many variables, Microsoft BASIC allows you to use all the letters of the alphabet PLUS it allows you to add a SECOND letter OR number to a variable to distinguish it from another. A1, X2, YY, Y2, and FF are legal variables. 1A, 22, or 3Z are not legal. Can you see why? Right, they do not begin with a letter of the alphabet! (2 letters or 1 letter and 1 number are max length allowed)

You may also assign a value to a variable that is assigned to other variables. For instance:

Press ENTER for an example of variable assignment?

```
10 X = 4
20 Y = 7
30 Z = X+Y
```

The variable Z is assigned the value of X+Y or 11.

All variables are assigned the value of 0 when you first start up Microsoft BASIC. However, some languages assign indefinite values to all variables at first, and wait for you to change them. That is why you may see programmers setting a variable to 0 when there appears to be no other reason for it.

Press ENTER?

Numeric Variables (cont)

When you use variables on the right side of an equation you must have assigned values to the variables previously. It's a KEY CONCEPT that the equal sign does not mean mathematical equality. The equal sign is an ASSIGNMENT statement. It ASSIGNS the value on the right side of the equation to the variable on the left.

```
10 X = X+1
```

In the above statement, X will be assigned the value of "X+1, or 1

Press ENTER?

Numeric Variables (cont)

10 X = 4

In the above example, we assigned the value of 4 to X.
In some dialects of BASIC, we must use the word LET to assign
a value to a variable.

10 LET X = 4

Such as above. It is not necessary to use the word LET in
Microsoft BASIC. We only mention it because you may wish to
copy a program written in another dialect onto Microsoft. If
you do, you may either leave the LET word in or drop it, the
BASIC language processor will accept either version.

press ENTER

Which of the following is a legal statement in Microsoft BASIC?

A 10 LET X1 = 2
B 20 1X = 2+3
C 15 XX2 = 56
D 10 2Z = X+Y

press the letter opposite the correct answer and press ENTER? A

CORRECT - Good job!

press ENTER

Which of the following is a legal statement if variables A and
B have previously been assigned a value?

A 10 A+B = C
B 10 C = A+B
C 10 ZZ = A+B
D 10 222 = A+B

press the letter opposite the correct answer? B

CORRECT - that was a KEY concept. you're doing good!

press ENTER?

Which do you want to do?

A Continue on
B Review this lesson again

press the letter opposite your choice and press ENTER? A

String Variables

Whenever you assign CHARACTERS (or NUMBERS that will not be mathematically manipulated - such as a street address) to a variable, you have created a STRING variable.

There is a special way of making a STRING in BASIC. You MUST attach a dollar sign ($) to the end of a variable label. When you do, the computer will know that this is a STRING and will not try to manipulate it. In addition, everything that you want to be included in the string must be enclosed in quotes. For example:

X$ = "The author is me"

press ENTER?

String Variables (cont.)

X$ = "The author is me"

Here, the variable label X is identified as a STRING variable by the addition of a dollar sign. Further, the CHARACTER data "The author is me" is assigned to the STRING.

X1$ = "114 West Cottage Street"

In this example, we have assigned both letters and numbers to
the STRING.

Examples of illegal STRING labels would be X, 1X$, X22$, JIM$

press ENTER?

Which of the following are correct STRINGS?

A A7$ = 12
B A7$ = "12"
C A7$ = North State Street
D H$ = North State Street

press the letter opposite the correct answer and press ENTER? B

CORRECT - are you sure you are only a student?

press ENTER?

The following program is an example of a BASIC program and its
output:

```
10 A$ = "This is really awesome. I mean really."
RUN

This is really awesome. I mean really.
```

Do you see how the computer treated the data?
What would be the output of this program:

```
10 Z2$ = "THIS IS A TEST QUESTION"
RUN

Type in the correct answer exactly as it would be printed
? THIS IS A TEST QUESTION
```

CORRECT

Press ENTER

Which do you want to do?

A Continue on
B Review this lesson again

Press the letter opposite your choice and press ENTER? A

Using Arithmetic

BASIC will let you use arithmetic to figure out almost any mathmatical task you would want. BASIC uses five symbols to represent addition, subtraction, multiplication, division and exponentiation (raising something to a power). Here they are:

Symbol	Meaning	Example
+	addition	A+B
-	subtraction	I-1
*	multiplication	3*2
/	division	3/2
	exponentiation	A^2 (A squared)

(Parenthesis -( )- are also used, just as in algebra)

Press ENTER?

Using Arithmetic (cont)

Note that a "*" always must be used for multiplication.
If you tried to use an X or x, the computer
would think you were trying to put in another variable label
and would give you an error message. Also, you cannot use
terms like I(5) to mean I*5. If you do, you will get an error.

If you put two variables together, like A and B, to make AB, you are not multiplying them, rather, you just created a NEW variable 'AB'!

So always remember to use the asterisk for multiplication.

press ENTER

Using Arithmetic (cont)

The symbols we just looked at are called ARITHMETIC OPERATORS and they may be combined in any order in a BASIC statement. However, just like mathematics, the computer will treat some symbols with a higher priority than others. For example:

X = 10+2/5-(2*3+2^2)

In this statement, the computer will scan the line and do all terms within parenthesis first. Then it will scan for exponentiation, perform those operations, then it will scan for multiplication OR division and perform those operations as it comes to them, and finally, it will scan for addition OR subtraction and perform those operations.

press ENTER

Using Arithmetic (cont)

The computer always scans from left to right. It will scan once for each category of symbols. The categories are restated below.

Category	Priority
( )	HIGHEST
^	Next HIGHEST
* or /	Next HIGHEST
+ or -	LOWEST

press ENTER for more

Using Arithmetic (cont)

X = 10+2/5-(2*2+2)^2

On the first scan, the computer would do the terms within the parenthesis. It would first do exponentiation (2^2 is 4, and then it would do the multiplication, and finally the addition. The value inside the parenthesis would be set at 10. Then it would do the terms outside the parenthesis in order of importance. First it would do the division, then it would do the addition (because it is scanning from left to right) and the subtraction last. Finally, it would set the value of X at 4.

press ENTER

Using Arithmetic

Parenthesis can be used to establish precedence within a statement. Suppose you want to make sure that the LAST part of a statement is calculated FIRST. You can use parenthesis. For example:

G = 22 (2+1)

is much different than:

G=22 2+1

Do you see why? The first value assigned to G is 75.27, the second value assigned is 11.5. Study the example carefully, press ENTER

Which of the following statements will assign the value of 1 to the variable M?

```
A  M = 1+2(2^2)-2
B  M = 5*2+(3*3+1)-10
C  M = 20/2(3*3+1)
D  M = 10*3-2

press the letter opposite the correct answer and press ENTER?

WRONG - the right answer is B (M = 5*2+(3*3+1)-10
                               (M = 5*2+(10)-10)
                               (M = 10 + 10 - 10)

press ENTER to continue?



Which do you want to do?

A  Continue on
B  Review this lesson again

press the letter opposite your choice and press ENTER? A
```

Comparing Variables

BASIC uses symbols to compare values to determine relationships such as whether one variable is less than, more than, or equal to another variable. We have already used one of these symbols it is called the equal sign (=). When you start programming, you will often want to check to see if one variable is different than another. There are six symbols you can use to do this.

press ENTER for examples?

Comparing Variables (cont)

(symbol table)

Symbol	Meaning	Example
=	equal	A=2
<	less than	A<B
>	greater than	A>B
<>	not equal to	A<>B
<=	less than or equal	A<=B
>=	greater than or equal	A>=B

We will discuss these in more detail in a later lesson.

press ENTER?

That concludes this lesson. When you hit ENTER you will be returned to the start of this part. You may either take the test or review selected areas.

By now, you should understand the following program.

```
10 PRINT"The product of 10 times 3 is "
20 PRINT 10 * 3
30 PRINT
40 PRINT
50 END
RUN
```

Your homework assignment will require you to write a program similar to this. Line 10 prints the string, line 20 prints the mathematical calculation. Lines 30 and 40 print two blank lines. You make the program work by using the RUN word after you have entered the statements. Your actual homework assignment is at the end of the test.

press ENTER?

LESSON 1B

This is the second part of a two part lesson It is divided into the following sections.

1) Library Functions 4) String Variables
2) Variables (general) 5) Using Arithmetic
3) Numeric Variables 6) Comparing Variables & Lesson Summary
7) TEST

A I'm taking this part in its entirety.
B I wish to review selected areas (or take the test).
C I want to go to the first part.
D I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? B

LESSON 1B

This is the second part of a two part lesson
It is divided into the following sections.

1) Library Functions 4) String Variables
2) Variables (general) 5) Using Arithmetic
3) Numeric Variables 6) Comparing Variables & Lesson Summary
7) TEST

Please type in the number beside the area you wish to review (1 through 7) and then press ENTER - press 0 and press ENTER to return to the Menu.

What is your choice? 7

FINAL TEST (lesson 1)

This test consists of 10 questions. You must get 70 percent of them correct to pass. (that's 7 right out of the 10 questions). Use only capital letters in your answers, don't include extra spaces or letters. If you answer a question wrong, you get the correct answer, plus a reference for review. In addition, you will get a synopsis of areas for review

at the end of the test.

If you successfully pass the test, you will be given your homework assignment. GOOD LUCK!

press ENTER to continue?

Is a computer program called Software?

A Yes
B No

press the letter opposite the correct answer and press ENTER? A

CORRECT

press ENTER?

Which of the following is an example of a FUNCTION

A LIST
B NEW
C SQR
D ADD

press the letter opposite the correct answer and press ENTER? C

CORRECT

press ENTER to continue?

Which statement would print the word TEST

A PRINT TEST
B PRINT "TEST"
C OUTPUT "TEST"
D PRINT 'TEST'

***** Listing of Program 'LESSON1.TXT' *****

press the letter opposite the correct answer and press ENTER? B

CORRECT

press ENTER?

What would the following program's output be?

10 X = SQR(4)
20 PRINT 5*X
RUN

Type in your answer and press ENTER? 10

CORRECT

press ENTER?

Give the necessary statement to print a blank line. Use
line number 10 and leave one blank space between terms.

What's your answer? 10 PRINT

CORRECT

press ENTER?

The two types of FUNCTIONS are LIBRARY and COMPUTER. TRUE or
FALSE?

A TRUE
B FALSE

press the letter opposite the correct answer and press ENTER? B

CORRECT

press ENTER?

Which of the following statements is invalid?

A 25 = X
B PRINT SQR(25)
C PRINT 25^10
D L = M+N

Type in the letter opposite the INCORRECT statement? A

CORRECT

press ENTER?

A string variable is made of mathematical equations which
will be manipulated by the computer and saved in temporary
memory.   TRUE or FALSE?

Type in TRUE or type in FALSE for this statement? FALSE

CORRECT

press ENTER?

Which of the following is an example of a peripheral device?

A CPU
B MEMORY
C KEYBOARD

press the letter opposite the correct answer and press ENTER? C

CORRECT

press ENTER?

If you had the following program in memory:

```
10  G=24
20  X=40
30  Y=10
40  L=3
```

What statement would you use to remove the middle two lines?

A NEW
B PRINT 10 + 40
C DELETE 20 to 30
D DELETE 20-30

press the letter opposite the correct answer and press ENTER? D

CORRECT

press ENTER?

You have finished the test. out of 10 possible correct answers
you scored 10 .

YOU HAVE PASSED

Do you want your score recorded on a permanent file?

A YES
B NO

Which? A

To record your score, we must open a file and put your name
in it. Therefore, surprisingly, we need your name. If your
name is not unique among the students likely to take this test,
please contact your test monitor for an identifying word that

will make you unique. Then enter that word below.

If you have already entered a score previously, be sure to
enter the same name you used before. (use all capitals)

ENTER your word or name now? DAN CREAGAN

You may now take LESSON 2. You will be returned to the MENU
from where you may go to LESSON 2 or quit.

press ENTER to return to MENU?

Break in 4180
Ready
SYSTEM*RESET #DO'

```
TRSDOS Ready
basic
BASIC 01.00.00 for TRSDOS Version 2
Copyright (c) 1983 By Microsoft, Licensed to Tandy Corporation.
All rights reserved.

Ready
run"lesson2



LESSON:  BASIC 2            VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON:  About One Hour


AUTHOR:  Capt Danny J. Creagan
         Air Force Institute of Technology

OBJECTIVE:  To teach the student about permanent storage,
            how to handle data in BASIC, and how to branch to
            different parts of a program.




press the ENTER key to continue?



                    LESSON 2

This is the first part of a two part lesson
It is divided into the following sections.

1)  Introduction          4)  REMarks
2)  Filenames             5)  INPUT Statements
3)  SAVE, LOAD, RUN


A  I'm taking this part in its entirety.
B  I wish to review selected areas.
C  I want to go to the second part.
```

D I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A

Do you wish to see an answer to the homework problem (Y/N)? Y

Here is one possible way to complete your homework:

```
10 PRINT"THE SUM OF 3, 2, AND 22 IS"
20 PRINT 3 + 2 + 22
30 PRINT
40 PRINT
RUN
THE SUM OF 3, 2, AND 22 IS
 27
```

You can try this one if you had trouble with yours. Press ENTER?

Introduction

In this lesson we will cover some of the most exciting and useful commands in the BASIC language. When we left off, in lesson 1, we had discussed some of the fundamental commands that you must use just to get BASIC started. Now, we will discover how to SAVE our programs for future use, how to control data input in our program, and how to leave little messages in our program so that other programmers can understand what we are trying to do. Most importantly, we will discover how to branch to different parts of a program depending on our data manipulation requirements. That way, one program can be extremely flexible and do many different kinds of work for us.

press ENTER to continue?

Introduction

After taking this lesson, we recommend you practice some of the techniques you have learned. The best way is to write a short program of your own and get it working. Then get a short program from a magazine or book and type it in. Don't be afraid to experiment with it. The best way of learning BASIC is to practice it.

press ENTER?

Filenames

Remember, in lesson 1, when we discovered how to make programs that could be RUN over and over? We said then that the program was stored in TEMPORARY memory. If you tried a few of the examples that were given, you will have noticed that the program was destroyed whenever you left BASIC. This section and the next section will show you how to SAVE a program, and then call it back from PERMANENT storage. That way, when you've spent hours making the best data manager ever written, you won't have to re-write it when you turn on the machine again!

PERMANENT storage is the way we store data for an indefinite period. We usually use DISKS or TAPE for PERMANENT storage.
press ENTER?

Filenames (cont)

For the purposes of this lesson, we will assume you only use DISKS for permanent storage.

A DISK is a platter of iron-oxide coated material that stores data almost the same way that an audio tape stores music. A DISK comes in many sizes and with many different storage capabilities. Fortunately, the way we store data on disk when we are using Microsoft BASIC is standardized for almost all

installations. (there is a slight difference if you are using a TRS-80, we will explain it as we go along)

press ENTER?

Filenames (cont)

What happens when you store data? Well, the computer takes care of most of the details. it waits until you tell it to store a program. then it searches the available storage areas to see if there is room for storage of your masterpiece. then it writes your data on the DISK. Remember, there may be MANY programs stored on a disk; therefore, each program must have a label that distinguishes it from the others. That way, the computer can find your program when you ask for it again.

press ENTER?

Filenames (cont)

This label is called a FILENAME. FILENAMES are very strictly controlled by the computer. They must follow the following format EXACTLY.

TRS-80	CROMEMCO (or CPM)
(filename)/(extension)	(filename).(extension)

Notice that the only difference between a TRS-80 and Cromemco is that the TRS-80 has a slash, '/', between the filename and the extension, while the Cromemco has a period or dot, '.'

press ENTER?

Filenames (cont)

TRS-80	CROMEMCO

(filename) (extension) (filename).(extension)

In the two examples, (filename), is an alphabetical character string no longer than eight (8) characters. (extension) is a file extension name that is also an alphabetical character string. The extension must not be longer than three (3) characters. The extension is OPTIONAL but, if used, must follow the format EXACTLY. NUMBERS may be used in both filenames and extensions, but they must NOT be the FIRST letter

press ENTER

Filenames (cont)

Here are some examples of legal filenames for your computer

MYPROG/BAS	MYPROG.BAS
MRHAPPY/BAS	MRHAPPY.BAS
SWIMFIN2/BAS	SWIMFIN
GOODNESS	GOODNESS.BAS

Notice that the filenames do not have to make sense, just so they mean something to the programmer who made them. (it would be unwise to name your program something common, like TEST.BAS, because someone else has probably already used that name. If they have, you will destroy their program when you SAVE your program to disk. Each program name must be unique.

press ENTER

Filenames (cont)

SWIMFIN.BAS

Notice, in the above filename, the extension is BAS. This would normally indicate that the file is a BASIC file (you may have word processor files, machine language files, or a variety of others). A good tip is to always save your BASIC files with this extension. That way, when you read the disk directory, you can tell that you have to go to BASIC

to run any program that has the extension - .BAS.

press ENTER?

Filenames (cont)

This section is very important to you. It has shown you what filenames are and what legal filenames look like. In the future you will use them a lot. Be sure you understand the idea behind filenames before you continue. It would be a good idea to look them up in your operating manual for Microsoft BASIC. There are many rules that were not covered here, but the rules we covered will get you by for now.

press ENTER?

ENTER a "T" if using a TRS-80, or "C" if CPM or CROMEMCO? C

In the Cromemco system, which of the following would be considered a legal filename.

A XXXXXXXX.FIL
B 24lesson.BAS
C TEST-BAS
D LUNCHTALK

press the letter opposite the correct answer and press ENTER?

WRONG the correct answer is A (XXXXXXXX.FIL)

press ENTER?

Is the extension necessary for a filename to be legal?

A YES
B NO

press the letter opposite the correct answer and press ENTER?

WRONG - the correct answer is B (the extension is not needed)

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

SAVE, LOAD and RUN

At the beginning of the last section, we said we would discover how to SAVE our programs so we wouldn't have to keep typing them in all the time. Well, this is it. To SAVE your program, (let's say you called it MYPROG.BAS), all you do is:

1) Type in the program
2) Type SAVE "MYPROG.BAS"
3) Congratulate yourself on a good job!

Be sure to notice that the filename is enclosed in quotation marks. That is mandatory. If you don't enclose the name in quotes, the command will 'BOMB' (it will fail).

press ENTER?

SAVE, LOAD, and RUN (cont

There are other things you should be aware of before you try to SAVE a program. First, there should be enough room on the disk to hold the program. If you are using a CROMEMCO hard disk, you will probably not have any problem in this area. ASK YOUR SYSTEM OPERATOR for more information. If you are using TRS-80 small floppy disk, then go to the COMMAND mode by typing in CMD"S" and then typing in DIR :0 or DIR :1. (drive 1 is called :0 and drive 2 is called :1 , then watch the display. You will see the free space left on the disk and a DIRECTORY of the files on the disk. If you have over 20 grans, then you have enough room for almost any program. SEE YOUR OPERATING MANUAL FOR MORE DETAILS.

press ENTER

SAVE, RUN, and LOAD (cont)

Remember, if you have enough space, then just type in this:

SAVE "MYPROG.BAS" or "MYPROG/BAS"
if using a TRS-80)

Now that we know how to SAVE a program, how do we get it back so we can RUN it again? That's easy. Just type in:

RUN"MYPROG.BAS"

press ENTER

SAVE, RUN, and LOAD (cont)

RUN"MYPROG.BAS"

When you type in the command, the computer will load your program THAT YOU HAD PREVIOUSLY SAVED, and RUN it.

What if you just saved a piece of a program because you were tired, expecting to come back at a later date and add to it?

If you did that, then you wouldn't want to RUN the program. You would just want to LOAD the program and LIST it to be sure it was the right one, then add the line numbers you need to complete the program.

press ENTER?

SAVE, RUN, and LOAD (cont)

You would LOAD the program using the same format as for SAVEing and RUNning it. That is:

LOAD"MYPROG.BAS"

BE SURE TO SAVE THE PROGRAM AGAIN AFTER YOU MODIFY IT, BECAUSE ONLY A COPY OF THE OLD VERSION WILL BE ON THE DISK!

press ENTER?

Which of the following is the correct command to SAVE a program named FRITZ?

A SAVE FRITZ
B LOAD FRITZ.BAS
C SAVE MYPROG
D SAVE"FRITZ"

press the letter opposite the correct answer and press ENTER?

WRONG - the correct answer is D (SAVE"FRITZ")

press ENTER?

You have just gotten to BASIC and want to load a program you have been working on. The filename for the program is SQUEEZE. How would you get the program from permanent storage to temporary memory? Which of the following would you type in:

A SQUEEZE
B LOAD"SQUEEZE"
C LOAD"SQUEEZE.BAS"

press the letter opposite the correct answer and press ENTER?

WRONG - the correct answer is B (LOAD"SQUEEZE"

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

REMarks

Something we should start early in our programming life, is DOCUMENTATION of how a program runs. You can include statements within a program THAT WILL NOT BE TOUCHED BY THE COMPUTER and will add to the clarity of your program. That way, when you LIST *your program (or when another programmer does)*, you can read the reminders left behind and more fully understand the program.

Those statements are called REM statements (REMark statements).

press ENTER?

REMarks (cont)

The format for a REM statement is: LINE #) REM (REMARKS)

An example is:

```
10 REM "this is how a REMark statement is made

when the computer sees REM it ignores all data that follows

press ENTER"



Here is an example of REMarks in a program:

5  R = 2
10  REM  the variable X in the next line is approx equal to PI
20  X = 3.14
25  C = X*R**2
30  PRINT C
RUN

Which would give:

12.56

Notice that the REM statement was not printed

press ENTER"



10  PRINT "SWEETUMS is a girl."
20  REM   "SWEETUMS is a girl."
30  PRINT "Her real name is CANDY"
40  REM   "I love her"
RUN

The above program would print the following (TRUE OR FALSE):

SWEETUMS is a girl.
 Her real name is CANDY

"ice is TRUE or FALSE, whichever is correct"

WRONG - the correct answer is TRUE - REM statements are not
printed
```

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

INPUT Statements

We saw in the first lesson that DATA can be assigned to a variable using the equals "=" sign. For example:

```
10  X = 10
20  PRINT X
RUN
```

Gives us:

```
10
```

In this example, we assigned 10 to X in line number 10.

press ENTER?

INPUT Statements (cont)

It is also possible to assign data while the program is running! THAT IS HOW THIS PROGRAM ASKS YOU QUESTIONS. It then tests your answer to see if you were right.

The BASIC word that it uses to ask the question is called an INPUT Statement. It looks like this:

```
10  INPUT"press the correct letter, then press ENTER";A
```

press ENTER"

INPUT Statements (cont.

10 INPUT"press the correct letter, then press ENTER";I$
RUN

Gives us:

press the correct letter, then press ENTER?

Notice that a question mark is automatically inserted after the message is printed. When the question is answered, the letter that the student selects is assigned to I$, just as if we had assigned a value to it in an equals statement.
Also note that a semi-colon is placed after the text

press ENTER"

INPUT Statements (cont)

Here is another example:

10 A = 2
20 INPUT"Enter a number between 1 and 9";N
30 C = A*N
RUN

Gives us:
Enter a number betwe 1 and 9?
 if we ENTER a 5 the

100

press ENTER"

INPUT Statements (cont.

```
10  INPUT"ENTER a number between 1 and 7"N
```

We have learned then, that the INPUT statement allows you to ENTER data in a program while it is running. It does this by stopping the program and waiting for you to enter data. When you do, it sets the data equal to the variable on the end of the INPUT statement.

Between the message or prompt and the variable, you must place a semi-colon (look at example above).

press ENTER?

INPUT Statements

You may use the INPUT statement without using a prompt or text message. If you do, then you must NOT put in a semi-colon. For example:

```
10  PRINT"when you see a question mark, ENTER a 5"
20  INPUT N
RUN
```

Gives you:

```
when you see a question mark, ENTER a 5
?
```

press ENTER?

INPUT Statements (cont)

```
10 PRINT"When you see a question mark, ENTER a 5"
20 INPUT N
RUN

when you see a question mark, ENTER a 5
?
```

Notice now the INPUT statement prompt (question mark) is on the following line? If we hadn't included line 10, we wouldn't know what to do when we saw the question mark. That's why you will see the text included in an INPUT statement most of the time. However, both ways are used.

press ENTER?

INPUT Statement (cont)

```
10 INPUT"ENTER a number between 1 and 9";N
RUN
```

If we were to save this example, and run it at a later date, We would always be asked for a number between 1 and 9. 'N' would always be changed from zero to the number we give it.

The values we assign to variables using the INPUT statement are not stored as part of the program. They are only temporarily held until we leave BASIC. They are reset to zero when we rerun the program.

press ENTER?

INPUT Statements (cont)

We have seen examples of both STRING variables and NUMERIC variables and we have seen both used with INPUT statements.

If you try to ENTER string data into a NUMERIC variable, you will get a "?REDO" message. That means you tried to ENTER data that was not proper for a NUMERIC variable.

A problem develops when you think you are entering NUMERIC data and you ENTER it into a STRING. You will not get an error message. Remember, but the right kind of variable on the end of the INPUT statement you will be tested on this.

press ENTER?

INPUT Statements (cont)

You may ENTER data into more than one variable using only ONE INPUT statement. Just put a comma "," between the variables, and a question mark will be prompted for each variable. For example:

```
10 INPUT"ENTER three (3) numbers";A,B,C
RUN

ENTER three (3) numbers? 10
?? 22
?? 5

press ENTER
```

INPUT Statements (cont)

```
10 INPUT"ENTER three (3) numbers";A,B,C
RUN

ENTER three (3) numbers? 10
?? 22
?? 5
```

Notice how the computer keeps prompting you with question marks until it gets all of its data. This can be a very useful routine, especially when you are asking for coordinates or for some other paired data input.

press ENTER

INPUT Statements (cont)

```
10 INPUT"ENTER three (3) numbers";A,B,C
RUN
```

ENTER three (3) numbers? 10,20,5

Notice that we hastened our data input. In this case, by ENTERing all the data on one line, separated by commas.

You may choose either way of ENTERing the data. It makes no difference.

press ENTER?

An INPUT statement CAN assign a value to a variable while the program is running? (TRUE or FALSE)

ENTER the word TRUE or ENTER the word FALSE?

WRONG - an INPUT statement IS used for inputting data while the program is running

press ENTER?

What is the prompt that an INPUT statement ALWAYS gives?

A A question mark
B Two question marks
C Quotes
D The word INPUT?

ENTER the correct letter (either A,B,C, or D.)?

WRONG - the prompt that is ALWAYS given is a question mark

press ENTER?

Which of the following is a valid response to this statement:

?) INPUT A,B,C$

```
A  2,ten,15
B  2,10,15 North Elm
C  TWO,1,22
D  22,NONE,8

ENTER the letter opposite the correct response and press ENTER?

WRONG - the correct answer is B (2,10,15 North Elm)

press ENTER?




Which do you wish to do?

A  Continue on
B  Review this section again

press the letter opposite the correct answer and press ENTER? A




GOING TO SECOND PART - PLEASE STANDBY




                LESSON 2B

This is the second part of a two part lesson
It is divided into the following sections.

1)  READ, DATA and RESTORE     3)  IF Statements
2)  Branching Introduction     4)  GOTO Statements & Summary
                5)  TEST


A  I'm taking this part in its entirety.
B  I wish to review selected areas (or take the test).
C  I want to go to the first part.
D  I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A
```

READ and DATA Statements

In the first part of this lesson, we learned that the INPUT statement is very efficient for assigning data to a variable while the program is running. However, when we have many data points to assign to variables, we need a more efficient mode. Think of how tedious it would be if you had to write statements to assign 250 data points in a program! (It is not unusual to have 10 times 250 data points for large regressions or forecasting programs)

How do we handle such a huge workload? One way is to use READ and DATA statements

press ENTER?

READ and DATA (cont)

READ and DATA statements are used like the equals sign '=' is used, but they are much faster and more versatile. Also, READ and DATA statements OPERATE WITHIN THE PROGRAM INSTEAD OF INTERFACING YOU WITH THE PROGRAM.

READ and DATA are two separate statements, but they are ALWAYS used WITH each other. The READ statement assigns the data as the program runs, and the DATA statement holds the values to be assigned.

press ENTER?

READ and DATA (cont)

The format of the READ statement is:

(line number) READ (variable or variables)

An example of a READ statement that would read values into variables B1, A, and XX$ is:

10 READ B1,A,XX$

press ENTER?

READ and DATA (cont)

10 READ B1,A,XX$

Note that both numeric and string variables may be 'read'
An example of a DATA statement that would be read is:

20 DATA 22.15,"AIN'T she sweet?"

Notice that the DATA statement has a different line number, but it follows the same format as the READ statement. When these two statements are placed in a program, the variables B1, A, and XX$ would hold 22, 15, and "AIN'T she sweet?" respectively.

press ENTER?

READ and DATA (cont)

The two statements can appear anywhere in a program and in any order, but for clarity, programmers usually place the DATA statement after the READ statement. Also, the DATA statements are usually grouped with other DATA statements in the program. We do that because it makes it easier to figure out another program when there is an order to how the programmer entered his statements.

press ENTER?

Is the following statement TRUE or FALSE?

A READ statement reads values from a DATA statement,
and places the values in variables that are to the
right of the READ word.

ENTER the word TRUE or ENTER the word FALSE?

WRONG - the sentence is valid.

press ENTER?

READ and DATA (cont)

10 READ B1,A,XX$

20 DATA 22.5,"AIN'T she sweet?"

The variables in the READ statement, and the values in the
DATA statement are separated by commas, and the string is
enclosed in quotes. You cannot assign string data to a
numeric variable. If you do, your computer will throw up
(well, at the very least it will ALWAYS give you an ERROR
message.

press ENTER?

READ and DATA (cont.

```
10 READ B1,A,Y
20 PRINT B1,A,Y
30 PRINT "The average of these numbers is";
40 PRINT (B1+A+Y)/3
50 DATA 5,10,15
RUN
```

Gives us:

```
5      10      15
The average of these numbers is 10
```

WOW! Study this example for a moment. Note that the calcula-
tions were printed beside the message instead of below it.

press ENTER?

READ and DATA (cont)

The calculations were not printed on another line because of the semi-colon after the print statement in line 30.

```
10 READ B1,A,Y
20 PRINT B1,A,Y
30 PRINT "The average of these numbers is";
40 PRINT B1,A,Y
50 DATA 5,10,15
```

Line 10 read the data in line 50. line 20 PRINTed it. and line 30 printed the message. The calculations in line 40 were printed on the end of the message due to the semi-colon at the tail of line 30.

press ENTER?

READ and DATA (cont)

Let's look at it once more:

```
10 READ B1,A,Y
20 PRINT B1,A,Y
30 PRINT "The average of these numbers is";
40 PRINT (B1+A+Y)/3
50 DATA 5,10,15
```

Gave us:

```
5      10      15
The average of these numbers is 10
```

press ENTER?

Now for a neat example:

```
10 READ A1,B1,C1
20 PRINT A1,B1,C1
30 DATA 1,2
RUN
```

Gives us:

```
OUT OF DATA IN 10
```

The BASIC language processor gave us an ERROR message that indicates we didn't have enough data for the number of variables that we tried to READ.

press ENTER?

READ and DATA (cont)

```
10 READ A1,B1,C1
20 PRINT A1,B1,C1
30 DATA 1,2
```

This program will BOMB (fail) because it will try to find a non-existent data point for the variable C1. If there had been more DATA points than READ variables, the program would have worked just fine. The next frame has an example of this.

press ENTER?

READ and DATA (cont)

```
10 READ A1,B1
20 PRINT A1,B1
30 DATA 1,2,3,4,99
```

Gives us:

```
 1            2
```

Although there were more data points that could have been read, there were no more variables left to READ them, so the program stopped. This would not cause an ERROR message.

press ENTER?

Is the following program valid?

```
10 READ X,Y,Z
20 DATA 25,2,15,55,64
30 PRINT Z
```

A Yes
B No

Enter the letter opposite the correct answer?

WRONG - the program is valid

press ENTER?

What is the value of Z that will be printed out?

WRONG - the correct answer is 15. Z is the third
variable to be read, so the third data point
is put in it.

press ENTER?

READ and DATA (cont)

Suppose you want to READ the same data points into DIFFERENT variables? Or perhaps you are making a program that will deal a deck of cards out, and you want to start over when you get to 52. You can reset the DATA statements so that variables will be assigned old data points by using the RESTORE statement.

press ENTER?

The RESTORE statement resets the DATA statements. After a RESTORE command, the next variable that is READ will be assigned the value that is just after the first DATA word.

```
10 READ A1,B1
20 PRINT A1,B1
30 RESTORE
40 READ C1,D1
50 PRINT C1,D1
60 DATA 1,2
RUN

 1            2
 1            2
```

Study this example closely and then press ENTER?

READ and DATA (cont)

This has been an extra long section and, if you are new to BASIC, you probably are confused about some of the rules. Don't let that worry you. Get out your BASIC manual (it came with your computer), and, after the little quiz that is coming up, go back and review this section again. THEN PRACTICE the techniques once you are through with this lesson. It is very important that you start practicing what you are learning. Practice with the manual beside you, and don't be afraid to ask an experienced programmer when you are confused. Use the examples seen in this program, or make up your own. You are one step closer to being a BASIC programmer!

press ENTER?

Is the following statement TRUE or FALSE?

The RESTORE statement causes the READ/DATA combination
to reset to the first data point to the right of the
first DATA statement.

A TRUE
B FALSE

ENTER the letter opposite the correct answer?

WRONG - the RESTORE command DOES reset the READ/DATA pair
        to the first data point past the first DATA word.

press ENTER?

READ and DATA statements are used within a program, and they
do NOT stop the program so DATA can be entered.

A TRUE
B FALSE

press the letter opposite the correct answer and press ENTER?

WRONG - READ and DATA statements DO NOT halt the program, they
        must be used within the program.

press ENTER?

```
10 READ A1,B1
20 PRINT A1,B1
30 RESTORE
40 READ C1,D1,E1
50 PRINT C1,D1
60 DATA 342,34
```

The above program is ERROR free

A TRUE
B FALSE

ENTER the correct answer (either A or B)?

```
WRONG - LINE 40 tried to read more DATA than was available.
        even though the RESTORE command was used.

press ENTER?
```

```
Which do you want to do?

A  Continue on
B  Review this lesson again

press the letter opposite your choice and press ENTER? A
```

Branching Introduction

```
10 A = 0
20 A = A+1
30 IF A = 5 THEN GOTO 50
40 GOTO 20
50 PRINT A
60 END
```

There are two types of branches, and we will be studying them in the next two sections. They are CONDITIONAL branches, and UNCONDITIONAL branches. The above program has both kinds in it

Line 30 is CONDITIONAL and line 40 is UNCONDITIONAL. Can you see why? Study this for a moment and then press ENTER?

Branching Introduction

```
10 A = 0
20 A = A+1
30 IF A = 5 THEN GOTO 50
40 GOTO 20
50 PRINT A
60 END
```

Line 30 is CONDITIONAL because it will only GO TO line 50 if the CONDITION that A = 5 is satisfied. That is, control will only be transferred to line 50 if A = 5.

press ENTER?

Branching Introduction

```
10 A = 0
20 A = A+1
30 IF A = 5 THEN GOTO 50
40 GOTO 20
50 PRINT A
60 END
```

Line 40 is UNCONDITIONAL because it will ALWAYS GO TO line 20 When it is executed. There will be no choice made. Control will go to line 20.

press ENTER?

Branching Introduction

```
10 A = 0
20 A = A+1
30 IF A = 5 THEN GOTO 50
40 GOTO 20
50 PRINT A
60 END
```

Notice that A will not equal 5 until line 20 is executed 5 times. Therefore, until A = 5, the CONDITION in line 30 will NOT be met and control will NOT GOTO line 50. Instead, it will go to the next line which is UNCONDITIONAL GOTO line 20.

Study this carefully, and then press ENTER?

Branching Introduction

```
10 A = 0
20 A = A+1
30 IF A = 5 THEN GOTO 50
40 GOTO 20
50 PRINT A
60 END
```

The statement in line 20 has made a COUNTER out of the variable A. Everytime the line is executed, A is incremented by one. COUNTERS are very useful in BASIC and we will discuss them more in a future lesson. For now, try to understand how this program works, and it will help you immensely in the future.

press ENTER?

The two types of branching are:

A CONDITIONAL and UNCONDITIONAL
B COUNTER and CONDITIONAL
C COUNTER and GOTO
D IF and GOTO

press the letter opposite the correct answer and press ENTER?

WRONG - the correct answer is A (CONDITIONAL and UNCONDITIONAL)

press ENTER?

```
10 A = 0
20 IF A = 3 GOTO 50
30 A = A+1
40 GOTO 20
50 PRINT A
60 END
RUN
```

What would be the output of this program?

```
A  1
B  2
C  no output would come from this program
D  3

press the letter opposite the correct answer and press ENTER? 

WRONG - the correct answer is D (3)

press ENTER? 



Which do you want to do? 

A  Continue on
B  Review this lesson again

press the letter opposite your choice and press ENTER? A



                    IF Statements

IF Statements are decision makers in BASIC.  They test to see
IF a condition is met. IF it is. THEN they execute the commands
that follow them on the same line.

The IF statement causes the program to make comparisons between
values.  It is one of the most powerful commands in the BASIC
language.  You have already seen how it can be used in the
previous section.  In this section, we will explain it in a
little more detail.

press ENTER? 



                  IF Statements (cont)

10 INPUT "ENTER a number between 1 and 10 (0 to quit)";N
20 IF N = 0 THEN STOP
```

```
30 IF N > 10 THEN PRINT "ERROR - you entered an invalid number"
40 IF N < 0 PRINT "ERROR - you entered an invalid number"
50 GOTO 10
```

This example shows three variations of the IF statement. If we RUN this program, and ENTER a '0' when prompted by line 10, then the CONDITIONAL statement in line 20 will be satisfied and the program will stop.

press ENTER?

IF Statements (cont)

```
10 INPUT "ENTER a number between 1 and 10 (0 to quit)";N
20 IF N = 0 THEN STOP
30 IF N > 10 THEN PRINT "ERROR - you entered an invalid number"
40 IF N < 0 PRINT "ERROR - you entered an invalid number"
50 GOTO 10
```

If we ENTER a 12 when prompted by line 10, the CONDITIONAL statement in line 20 will not be satisfied, nothing will happen until control passes to line 30. At that time, the check for N GREATER THAN 10 will be met, and the message will be printed. No other condition will be met until control gets to line 50. The UNCONDITIONAL GOTO on line 50 will send control back to the beginning of the program.

press ENTER to continue?

IF Statements (cont)

```
10 INPUT "ENTER a number between 1 and 10 (0 to quit)";N
20 IF N = 0 THEN STOP
30 IF N > 10 THEN PRINT "ERROR - you entered an invalid number"
40 IF N < 0 PRINT "ERROR - you entered an invalid number"
50 GOTO 10
```

If we ENTER a -12 when prompted by line 10, we will satisfy the CONDITIONAL statement in line 40, the message will be printed and control will eventually get back to line 10.

press ENTER?

* IF Statement (cont)

The IF statement can also be used to compare two expressions such as:

30 IF (20*2-3) < (3*10+6) THEN GOTO 190

Also, variable assignment can be done in an IF statement:

30 IF (20*2-3) < (3*10) THEN A=1

press ENTER?

There is another word that can be added to the IF statement to make it more powerful. It is the ELSE word.

220 IF A = 1 THEN GOTO 10 ELSE GOTO 200

In this line, if the variable A equals 1 then control transfers to line 10, if it does NOT equal 1 then control transfers to 200. In this case, *something* ALWAYS happens at line 220 because of the ELSE statement.

220 IF A = 1 THEN GOTO 10 ELSE IF A = 2 GOTO 30

In this case, if A = 1 or A = 2 then something will happen in line 220, if none of the CONDITIONS are met, then the line will not be executed.

press ENTER?

IF Statement (cont)

IF (true/false expression) THEN (action) ELSE (action)

The IF statement instructs the computer to test the following logical or relational expression. If the expression is TRUE then control will proceed to the action line after the THEN word. If the expression is not true, then control will proceed to the ELSE action.

press ENTER?

What is the output of the following program?

```
10 A = 255
20 IF A < 190 THEN PRINT "TOO WEAK"
30 IF A > 254 THEN PRINT "TOO STRONG"
40 IF A = 255 THEN PRINT "A = 255"
```

A TOO WEAK
  A = 255
B TOO STRONG
  END
C TOO STRONG
  A = 255
D A = 255

ENTER the letter opposite the correct answer?

WRONG - the correct answer is C (TOO STRONG,
                                (A = 255)

press ENTER?

The IF statement is a CONDITIONAL statement.

Is the above sentence TRUE or FALSE?

A TRUE
B FALSE

choose the letter opposite the correct answer and press ENTER?

WRONG - the IF statement IS a CONDITIONAL statement.

press ENTER?

Which do you want to do?

A Continue on
B Review this lesson again

press the letter opposite your choice and press ENTER? A

GOTO Statements

Conditional branches are written in programs with IF THEN ELSE statements. Unconditional branches are written with GOTO statements.
As we saw earlier, GOTO directs control of a program to another line. For example:

```
10 INPUT "ENTER a number between 1 and 10";N
20 IF N = 8 THEN GOTO 50
30 IF N <> 8 THEN PRINT "GUESS AGAIN"
40 GOTO 10
50 PRINT "YOU GUESSED IT
60 END
```

press ENTER?

```
10 INPUT "ENTER a number between 1 and 10";N
20 IF N = 8 THEN GOTO 50
30 IF N <> 8 THEN PRINT "GUESS AGAIN"
40 GOTO 10
50 PRINT "YOU GUESSED IT
60 END
```

The GOTO statement in line 40, when executed, sends control to the beginning of the program.

press ENTER?

GOTO (cont)

You can make the GOTO statement a MULTI-way branching statement by modifying it slightly. For example:

```
10 INPUT "ENTER a number between 1 and 3";N
20 ON N GOTO 30,50,70
30 PRINT "YOU ENTERED A ONE (or an illegal number)"
40 STOP
50 PRINT "YOU ENTERED A TWO"
60 STOP
70 PRINT "YOU ENTERED A THREE"
80 STOP
```

press ENTER?

```
10 INPUT "ENTER a number between 1 and 3";N
20 ON N GOTO 30,50,70
30 PRINT "YOU ENTERED A ONE (or an illegal number)"
40 STOP
50 PRINT "YOU ENTERED A TWO"
60 STOP
70 PRINT "YOU ENTERED A THREE"
80 STOP
```

When line 20 is executed, the value of N is used to count over 'N' elements passed the GOTO word. Control branches to the line number indicated by this 'Nth' element. If there is no element that corresponds to the value of N, then control passes to the next available line. Press ENTER when ready?

```
10 INPUT "ENTER a number between 1 and 3";N
```

```
20 ON N GOTO 30,50,70
30 PRINT "YOU ENTERED A ONE (or an illegal number)"
40 STOP
50 PRINT "YOU ENTERED A TWO"
60 STOP
70 PRINT "YOU ENTERED A THREE"
80 STOP
```

The value of N MUST be greater than 0 and less than 255. If it is not, BASIC will print an error. If N is 1, 2, or 3 then the program will print the appropriate message and stop.

press ENTER?

Which lines (beyond 20) are executed if you ENTER a 10 here?

```
10 INPUT "ENTER a number between 1 and 3";N
20 ON N GOTO 30,50,70
30 PRINT "YOU ENTERED A ONE (or an illegal number)"
40 STOP
50 PRINT "YOU ENTERED A TWO"
60 STOP
70 PRINT "YOU ENTERED A THREE"
80 STOP
```

A 30 and 40
B 50 and 60
C 70 and 80

press the letter opposite the correct answer then pres ENTER?

WRONG - the correct answer is A (30 and 40)

press ENTER?

You are now done with this lesson. When you hit ENTER, you will be returned to the MENU where you may review sections or take the TEST.

Remember, after you are done here, practice some of the

things you have learned. And keep a BASIC manual by your side.
This program will show you the fundamentals, you have to teach
yourself how to be good at BASIC. That means you must PRACTICE

press ENTER to go to the MENU?

LESSON 2B

This is the second part of a two part lesson
It is divided into the following sections.

1) READ, DATA and RESTORE 3) IF Statements
2) Branching Introduction 4) GOTO Statements & Summary
5) TEST

A I'm taking this part in its entirety.
B I wish to review selected areas (or take the test).
C I want to go to the first part.
D I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? B

LESSON 2B

This is the second part of a two part lesson
It is divided into the following sections.

1) READ, DATA and RESTORE 3) IF Statements
2) Branching Introduction 4) GOTO Statements & Summary
5) TEST

Please type in the number beside the area you wish
to review (1 through 5) and then press ENTER - press 0 and
press ENTER to return to the Menu.

What is your choice? 5

FINAL TEST (lesson 2)

This test consists of 10 questions. You must get 70 percent or them correct to pass. (that's 7 right out of the 10 questions). Use only capital letters in your answers, don't include extra spaces or letters. GOOD LUCK

press ENTER to continue?

Which of the following is a legal filename?

A SCOIXIE.CMD (in CPM or Cromemco)
B TRIUMPH/650 (in TRS-80)
C THEWAYOF.B21 (in CPM or Cromemco)
D SCORE?/DAT (in TRS-80)

ENTER the letter opposite the correct answer?

WRONG - the correct answer is C
Answers A and B filenames or extensions start with numbers instead of alphabet characters. Answer D has a non alpha-numeric character in it.
See part 1. filenames. in lesson 2.

press ENTER?

Which of the following commands will load a file called 'LOVE'?

A CREATE "LOVE"
B RUN "LOVE"
C LOAD "LOVE.BAS2"
D SAVE "LOVE"

ENTER the correct answer?

WRONG - the correct answer is B
In answer A. CREATE is not a BASIC word. in C

an incorrect extension was used (no extension was
needed), in D the program would be saved, not loaded.
See part 1. SAVE, LOAD, RUN of lesson 2.

press ENTER?

REMark statements are similar to PRINT statements, except that
the computer takes less time to print them.

A TRUE
B FALSE

ENTER the letter opposite the correct answer?

WRONG - the correct answer is B
REMark statements are not output to the screen, they
are only used for programmer information. See part 1

press ENTER?

Which of the following examples is INVALID?

A INPUT T$
B INPUT"ENTER YOUR NAME";N
C INPUT N
D INPUT"ENTER YOUR AGE";N

ENTER the letter opposite the correct answer?

WRONG - statement B is the bad one because it tries
to load a numeric variable with string data.
See Part 1.

press ENTER?

What command will let you put your program into permanent
storage so that you can recall it later?

```
A  SAVE
B  RUN
C  LOAD
D  STORE

ENTER the letter opposite the correct answer?

WRONG - the correct answer is A
        RUN causes program execution. LOAD loads the
        program from disk and STORE is not a BASIC word.
        See part 1.

press ENTER?



Which of the following statements is legal?

A  READ A$ B$ C1
B  DATA A B C
C  READ 22.33.44
D  DATA "QUIT","FIRE"

ENTER the letter opposite the correct answer?

WRONG - the correct answer is D
        Answers A and B don't have commas between variables.
        and answer C tries to use constants instead of
        variables for the READ.
        See part 2. READ and DATA.

press ENTER?



READ and DATA statements halt the program so the operator can
insert correct answers.

A  TRUE
B  FALSE

ENTER the letter opposite the correct answer?
```

```
WRONG - the correct answer is B
        READ and DATA are used to load variables WITHOUT
        stopping the program.
        See part 2, READ and DATA.

press ENTER?



The IF statement is a CONDITIONAL BRANCHING statement.

A  TRUE
B  FALSE

ENTER the letter opposite the correct answer?

WRONG - the correct answer is A
        See Part 2, IF and GOTO.

press ENTER?



The following program will NOT have an output - TRUE or FALSE?

10 N = 5
20 ON N GOTO 40,50,30
30 STOP
40 PRINT N
50 STOP
60 PRINT N
70 STOP
80 PRINT N
90 STOP

A  TRUE
B  FALSE
ENTER the letter opposite the correct answer?

WRONG - the correct answer is A
        Line 10 sets N to 5. line 20  only has 3 places
        to go to. so it defaults to the line under it.
        That line is a STOP statement. See part 2. GOTO.
```

press ENTER?

The following program will have an output - TRUE or FALSE?

```
10 N = 3
20 IF N = 3 THEN GOTO 40
30 PRINT N
40 N = 4
50 END
```

A TRUE
B FALSE

ENTER the letter opposite the correct answer?

WRONG - the correct answer is B
Line 10 sets N to 3. line 20 causes the program to
go to line 40. then 50.
See part 2. IF.

press ENTER?

You have finished the test. out of 10 possible correct answers
You scored 0 .

YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:

part 1. Filenames
part 1. SAVE, LOAD. RUN
part 1. REMarks
part 1. INPUT Statements
part 2. READ and DATA Statements
part 2. IF Statements
part 2. GOTO Statements

press ENTER?

```
YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS

YOU SHOULD RETAKE LESSON 2!

You will be returned to the Menu.


press ENTER to return to the MENU?

Break in 3330
Ready
SYSTEM"REGEE *DD"
```

```
TRSDOS Ready
basic
BASIC 01.00.00 for TRSDOS Version 2
Copyright (c) 1982 By Microsoft. Licensed to Tandy Corporation.
All rights reserved.

Ready
RUN"LESSON3
```

LESSON: BASIC 3 VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON: Less than one hour

AUTHOR: Capt Danny J. Crescan
Air Force Institute of Technology

OBJECTIVE: To teach the student how to use LOOPS and ARRAYS.

press the ENTER key to continue?

LESSON 3

This is the first part of a two part lesson
It is divided into the following sections.

1) Introduction & FILL 4) FOR NEXT Statements
2) LOOPs (Intro) 5) Advanced FOR NEXT
3) COUNTER variables

A I'm taking this part in its entirety.
B I wish to review selected areas.
C I want to go to the second part.

D I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A

Introduction & KILL Statement

By now you should have saved a few small programs to disk, and you should have practiced all the commands we have discussed to date. If you have wondered how to get rid of a file that you were done with, or have mistakenly saved and didn't need one of your programs, the next frame will be of service to you. It describes the KILL statement. We have purposely put the KILL statement in the third lesson because it is a dangerous command that can eliminate valuable and irreplacable data or programs if used incorrectly. By now you should feel comfortable with some of the commands and you shouldn't make the mistake of KILLing someone else's files, or your own by accident. Remember, KILL does just what it says, it KILLs files

press ENTER to learn about the KILL command

Introduction & KILL (cont)

The format for the KILL command is:

KILL "<filename.ext"

You must enclose the filename in quotation marks.
The extension is only needed when the original file had one.

Once this command is entered, the file will be removed (unless protected by passwords - you can learn about passwords from your system manual). If the file is removed, there is very little chance of recovering any of your data. In some cases an advanced programmer can retrieve data from a killed file but the process is difficult and often fails. BE CAREFUL

press ENTER

Which of the following statements is legal?

A KILL STUPID
B KILL "STUPID.ICO"
C KILL "STUPID.BAS"

press the letter opposite the correct answer and press ENTER? C

CORRECT

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

LOOPs (Intro.

One of the most exciting aspects of computer programs is their ability to patiently check and recheck data, and to tirelessly calculate figures and columns of numbers. (the exciting part is that YOU don't have to spend hours doing drudgery when the computer can spend minutes or seconds doing the same job. One of the tasks of the programmer is to efficiently use his computer memory to program the time consuming tasks.

press ENTER?

Suppose you wanted to display 3 columns of figures, the left column would be integers from 1 to 8, the middle column would be the square of the figure in the adjacent first colum.

and the third column would be the square of the figure in the adjacent middle column. It would look like this:

1	1	1
2	4	16
3	9	81
4	16	256
5	25	625
6	36	1296
7	49	2401
8	64	4096

press ENTER?

1	1	1
2	4	16
3	9	81
4	16	256
5	25	625
6	36	1296
7	49	2401
8	64	4096

If you were to write separate print statements for these calculations, you would have to write at least 9 lines of code. Using LOOPs, you can calculate this column of figures, and display it, in three lines of code!

In the next sections we will learn this and more...press ENTER?

COUNTER variables

The first step in learning about loops is to understand COUNTER variables. A COUNTER is a variable that is used to keep track of the number of times a program executes a line. It is often used in conjunction with an IF THEN statement. For example:

```
1. I=1
2. I=I+1
```

```
30 IF I = 10 THEN STOP
40 GOTO 20
```

The value of I after RUNning this program is 10.

press ENTER?

COUNTERs (cont)

```
10 I=0
20 I=I+1
30 IF I = 10 THEN STOP
40 GOTO 20
```

The COUNTER variable is I. It is INCREMENTED ever. time line
40 sends control to it. When the CONDITIONAL statement in
line 30 discovers that I is equal to 10, it executes the STOP.
At the end of the program, I is equal to 10.

press ENTER?

```
10 I=1
20 I=I+1
30 IF I = 10 THEN STOP
40 GOTO 20
```

Do you see what would happen if we didn't include the IF
statement? The program would look like this:

```
10 I=0
20 I=I+1
30 GOTO 20
```

The program would never stop. It would be caught in an ENDLESS
LOOP. We have many terms for this common mistake. Some
examples: LOCKED UP, HUNG, and *** DUMMY, you did it again! ***

press ENTER?

COUNTERs (cont)

If your program ever gets hung up, you may have to type
(CONTROL) C (if CPM or Cromemco), or (BREAK) (if TRS-80)
In some extreme cases, you may have to reset the system.
On the TRS-80, the RESET button is the red recessed button on
the upper right of your keyboard (left rear if using a MODEL I)
CAUTION, if you are using this program on a multi-user system,
such as Cromemco System II, DO NOT RESET the system, let the
operator help you fix the problem. If you are using a single
user system, such as a TRS-80, then RESET will work as an exit
from the locked up program (but try (BREAK) first). If you use
either RESET or (CONTROL) C, the system will take you to the
COMMAND mode, and you will have to re-initialize BASIC.

press ENTER?

What is the COUNTER variable in this program?

```
10 Z=0
20 X=0
30 X=X+1
40 IF Z    X THEN Z = Z
50 IF X = 5 THEN STOP
60 PRINT Z
70 GOTO  30
80 END
```

A Z is the COUNTER
B X is the COUNTER
C BOTH are COUNTERs, but X will control the program

press the letter opposite the correct answer and press ENTER? B

CORRECT - that shows good understanding of the principle!

press ENTER?

which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

FOR - NEXT Statements

In the previous discussion of counters, we showed you that we tested a counter with an IF statement to see if it had reached a desired level. If it had, then we went to another part of the program, or STOPped. There is a set of statements in BASIC that lets us shorten the statements needed to duplicate this kind of program. The set of statements is called FOR - NEXT.

press ENTER?

FOR - NEXT (cont)

COUNTER loop

```
10 X=0
20 X=X+1
30 PRINT X
40 IF X = 5 GOTO 60
50 GOTO 20
60 END
```

FOR NEXT loop

```
10 FOR X = 1 to 5
20     PRINT X
30 NEXT X
40 END
```

The two programs above will print exactly the same output. The COUNTER loop requires more statements and is not as efficient as the FOR NEXT loop.

press ENTER?

COUNTER loop

```
10 X=0
20 X=X+1
30 PRINT X
40 IF X = 8 GOTO 60
50 GOTO 20
60 END
```

FOR NEXT loop

```
10 FOR X = 1 to 8
20     PRINT X
30 NEXT X
40 END
```

Line 10 of the FOR NEXT loop initializes X, the same way that lines 10 and 20 of the COUNTER loop do. The FOR NEXT loop sees line 10 as 'I'm going to start a loop. X is the counter. I will begin with X = 1 and when X is GREATER than 8 I will GOTO the statement that is after the NEXT X statement'.

press ENTER

COUNTER loop

```
10 X=0
20 X=X+1
30 PRINT X
40 IF X = 8 GOTO 60
50 GOTO 20
60 END
```

FOR NEXT loop

```
10 FOR X = 1 to 8
20     PRINT X
30 NEXT X
40 END
```

Line 30 of the COUNTER loop and line 20 of the FOR NEXT loop are the same and perform the same function.
Note that the PRINT statement was indented a few spaces in the FOR NEXT loop. This lets you see the loop structure better. You should do that in your own programming, too.

press ENTER

COUNTER loop FOR NEXT loop

```
10 X=0                          10 FOR X = 1 to 9
20 X=X+1                        20     PRINT X
30 PRINT X                      30 NEXT X
40 IF X = 9 GOTO 60             40 END
50 GOTO 20
60 END
```

The IF statement in line 40 of the COUNTER loop determines if X has reached 9. This was done automatically by the FOR NEXT loop because the last value in line 10 was specified as 9. Line 30 of the FOR NEXT loop is the same as the UNCONDITIONAL GOTO in line 50 of the COUNTER loop. The NEXT X statement INCREMENTS X and sends control to the FOR statement. In this case control goes to line 10.....press ENTER?

```
10 FOR X = (value #1) to (value #2)
20     PRINT X
30 NEXT X
```

In summary, the FOR X = (value #1) to (value #2) causes the variable X to be initially set at value #1. and the program executes the next lines until it reaches the NEXT X statement. (X is an example, any VARIABLE works). The NEXT statement causes an UNCONDITIONAL GOTO to the FOR statement. The variable is incremented and the FOR word tests to see if it EXCEEDS value #2. IF it does, then control passes to the statement that follows the NEXT statement. If there are no statements passed the NEXT word, the program ENDs.

press ENTER?

FOR NEXT (cont)

This concludes the initial FOR NEXT section. You should understand what a simple FOR NEXT statement does. If you do not, you will be given a chance to review this section before you go on. But before that - QUIT time!

press ENTER?

```
1
4
9
```

Which of the following programs made the above list?

```
A  10 FOR X=1 TO 3          C  10 FOR 1 = 1 TO 4
   20    PRINT X*2             20    PRINT 1*2
   30 NEXT X                   30 NEXT 1

B  10 X=X+1                 D  10 Z=Z+1
   20 print X*2                20 print Z*2
   30 IF X=3 THEN GOTO 10      30 IF Z=4 THEN STOP
   40 GOTO 10                  40 GOTO 10
```

press the letter opposite the correct choice and press ENTER? A

CORRECT

press ENTER?

The NEXT statement is the same as an UNCONDITIONAL BRANCHING Statement. It branches to the next line under the FOR statement.

Is the paragraph above TRUE or FALSE?

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

WRONG - the NEXT statement is UNCONDITIONAL, but it branches to the SAME line as the FOR statement. It also increments the variable before the FOR statement tests it.

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER" A

Advanced FOR NEXT

```
10 FOR X = 1 TO 10 STEP 2
20    PRINT X
30 NEXT X
```

Gives us:

```
 1
 3
 5
 7
 9
```

Interesting, isn't it? ..... press ENTER for an explanation?

Advanced FOR NEXT

```
10 FOR X = 1 TO 10 STEP 2
20    PRINT X
30 NEXT X
```

The STEP word in the FOR statement caused the value of X to be incremented by 2 instead of 1. The output of 1.3.5.7.9 was correct because the NEXT statement is the statement that actually increments the value, so the first time through, X was equal to 1, the next time through it was equal to 3, etc.

press ENTER"

Advanced FOR NEXT

```
10 FOR X = 10 to 1 STEP -2
20   PRINT X
30 NEXT X
```

Gives us:

```
 10
 8
 6
 4
 2
```

press ENTER for explanation?

```
10 FOR X = 10 to 1 STEP -2
20   PRINT X
30 NEXT X
```

In this case, we STEPped DOWN instead of up. Notice that the first value of the FOR statement is the largest value. It would not make sense to specify a negative STEP and give the range of a positive STEP. For example:

```
10 FOR X = 1 to 10 STEP -1
```

This doesn't make sense, and would not work.

press ENTER?

You may include a FOR NEXT statement within another FOR NEXT statement. If you do, it is called NESTED looping.
For example:

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         PRINT X,Y
```

```
40     NEXT Y
50 NEXT X
```

Before we explain the output, do you see how we indented the statements within the first loop, and then further indented the statements within the second loop? It makes them easier to understand (and explain).

press ENTER for output?

Advanced FOR NEXT (NESTED LOOPS)

```
 1             1
 1             2
 2             1
 2             2
```

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         PRINT X,Y
40     NEXT Y
50 NEXT X
```

Try to figure how the program produced the columns above it and then press ENTER?

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         PRINT X,Y
40     NEXT Y
50 NEXT X
```

On the first pass through the program, the FOR NEXT loop for X was set up, and control passed to the lines between FOR X = 1 to 2, and NEXT X. Those lines happened to be another loop with the variable Y as the FOR NEXT variable. When the Y FOR NEXT executed the first time, X was equal to 1 and Y was printed twice, once as a 1 and once as a 2. When the Y loop

finished, the NEXT X statement caused control to GOTO line
10 again. The process was repeated, but X now was equal to 2.

press ENTER?

```
 1             1
 1             2
 2             1
 2             2
```

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         PRINT X,Y
40     NEXT Y
50 NEXT X
```

Can you figure it out now? Think about it and then press ENTER?

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         PRINT X,Y
40     NEXT Y
50 NEXT X
```

The most common mistake that programmers make when using nested loops, is mislabeling the NEXT statements. Notice that the NEXT statement for the Y variable is placed before the NEXT X statement.

press ENTER?

The format for nesting loops is:

```
LOOP 1 (for)
      LOOP 2 (for)
            LOOP 3 (for)
```

```
            LOOP I (next)
      LOOP J (next)
LOOP K (next)
```

Notice that you must back out of a nest in reverse order.

press ENTER?

Is the following program valid?

```
10 FOR X = 1 to 200
20     FOR Z = 1 to 2
30         PRINT X+Z
40     NEXT Z
50 NEXT X
```

A YES it is a good program
B NO it is not a valid program

ENTER the letter opposite the correct answer?

WRONG - the correct answer is A. There is nothing wrong
        with the program.
press ENTER?

Give the first statement of a FOR NEXT loop, that will cause the loop to increment the variable I from one to 20 in steps of 2. Use 10 as your line number and leave one space between all words, numbers, and variables.

ENTER your answer?

WRONG - the answer is:
        10 FOR I = 1 TO 20 STEP 2

press ENTER?

Is the following program valid?

```
10 FOR X = 1 TO 2
20     FOR I = 1 TO 5
30         PRINT "This is the last question"
40     NEXT I
50 NEXT X
```

A Yes, it is valid
B No, it is not valid

ENTER the letter opposite the correct answer?

WRONG - the program is valid

press ENTER?

Remember in the beginning of this section when we mentioned the program that produced three columns of figures, the first column was integer 1 through 8, the second column was the square of the first, and the third the square of the second? Think you can figure out how we did it?

press ENTER for the answer?

1	1	1
2	4	16
3	9	81
4	16	256
5	25	625
6	36	1296
7	49	2401
8	64	4096

```
10 FOR X = 1 TO 8
20     PRINT X,X*X,(X*X)*(X*X)
30 NEXT X
```

It looks like child's play now, doesn't it?

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

Going to second half of lesson 3 - Wait one moment

LESSON 3B

This is the second part of a two part lesson
It is divided into the following sections.

1) Arrays (Introduction) 3) Dimension & Clear
2) Arrays   4) Test

A I'm taking this part in its entirety.
B I wish to review selected areas. (or take the test)
C I want to go to the first part.
D I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A

Arrays (Intro)

We have seen that variables are storage places for data. In large programs, it is difficult to manipulate large data bases without having numerous variables to assign the bits of data to. However, there is a way to group our variables into similar bunches that makes it easier for us to *tell* what part of the data base our variable belongs to. We can use

ARRAYs.

press ENTER?

Arrays (Intro)

One use for ARRAYs would be to make a training program that listed the people on training, their time in the organization, or rank, and their training status. We could group the major catagories (name, rank, training status) into three variables and use subscripts to provide a place for each entry in our data base.

press ENTER?

Name	Rank	Training Status
John Doe	Foreman	9
Jake Robinski	Peon	3
Mark Muffin	Specialist	5

We could assign subscripted ARRAYs to the three main catagories. An ARRAY has the following format:

    Variable (subscript)

The subscript is enclosed in parenthesis. Examples of valid ARRAY variables are: N$(1), R$(2), T(9)

press ENTER?

Name	Rank	Training Status
John Doe	Foreman	9
Jake Robinski	Peon	3

```
Mark Muffin              Specialist          5
```

We could use our ARRAYs to hold the above data.
We could use N$(0), N$(1), and N$(2) to indicate the three names, R$(0), R$(1), and R$(2) to indicate the three ranks, and T(0), T(1), and T(2) to represent the three training levels. Note that 0 is a valid subscript.

press ENTER?

```
10 N$(0) = "John Doe"
20 N$(1) = "Jake Robinski"
30 N$(2) = "Mark Muffin"
40 R$(0) = "Foreman"
50 R$(1) = "Peon"
60 R$(2) = "Specialist"
70 T(0)  = 9
80 T(1)  = 3
90 T(2)  = 5
100 PRINT"Name              Rank              Training Status"
110 PRINT
120 PRINT N$(0),R$(0),T(0)
130 PRINT N$(1),R$(1),T(1)
140 PRINT N$(2),R$(2),T(2)
```

This program, when RUN, would print our data.....press ENTER?

```
Name                      Rank            Training Status

John Doe                  Foreman               9
Jake Robinski             Peon                  3
Mark Muffin               Specialist            5
```

This is what that program would output. Notice that we have used only three variables, but we made them ARRAYs so that we could hold nine bits of data.
As we continue through our lessons, we will discover some very powerful uses for ARRAYs.

press ENTER?

Is the following ARRAY and its subscript valid?

A(0)

A TRUE
B FALSE

ENTER the letter opposite the correct answer? B

WRONG - the correct answer is A

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

ARRAYs

If you have a little mathematics in your background, you will have noticed that ARRAYs are almost the same as their math equivalent, except that the subscripts are in parenthesis instead of slightly lower and to the right of the variable.

The previous examples all dealt with a ONE-DIMENSIONAL ARRAY. That is, there was only one number in parenthesis that was significant. ARRAYs with TWO, THREE, FOUR, or more dimensions are possible. Most dialects of BASIC, including Microsoft, will handle at least 3 dimensions. An example of a TWO dimension ARRAY would be N(2,2). Notice that the extra dimension was designated by just adding another subscript in-

side the parenthesis. A THREE DIMENSIONED ARRAY looks like this: R(2,1,9) or T$(5,44,3) (or any combination of numbers) press ENTER?

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         READ A(X,Y)
40         PRINT A(X,Y);
50     NEXT Y
60 NEXT X
70 DATA 5,10,15,20
```

The above program combines several of the techniques that we have been learning. Before we tell you the answer, try to figure out what the output of the program would be. We warn you, it is a little tricky, but see if you can figure it out.

press ENTER?

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         READ A(X,Y)
40         PRINT A(X,Y);
50     NEXT Y
60 NEXT X
70 DATA 5,10,15,20

5 10 15 20
```

This is the output. Lines 10 and 20 are nested FOR NEXT loops. They set up the READ statement in line 30 so that it will READ in the values that are in the DATA statement and assign the current X,Y subscript to it. Values are read in one at a time. press ENTER?

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         READ A(X,Y)
40         PRINT A(X,Y);
50     NEXT Y
60 NEXT X
70 DATA 5,10,15,20
```

On the first pass, X = 1 and Y = 1. A(1,1) therefore, equals 5 The semi-colon on the end of the print statement causes the numbers to be printed side by side instead of on separate lines and they all have one space between them. (caused by the ';') On the second iteration of Y, Y will equal 2 and X = 1. A(1,2) will equal 10.

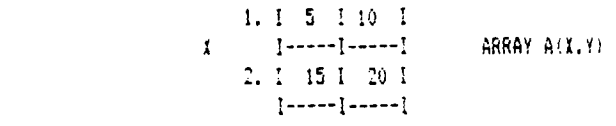
press ENTER?

```
10 FOR X = 1 TO 2
20     FOR Y = 1 TO 2
30         READ A(X,Y)
40         PRINT A(X,Y);
50     NEXT Y
60 NEXT X
70 DATA 5,10,15,20
```

When the second iteration of Y is done, control will pass to line 10 and X will begin ITs second iteration. The Y loop will start all over again and when line 30 is executed (3rd time), A(2,1) will equal 15. Finally, Y will execute for the fourth time (second time while X = 2), and A(X,Y) will have all numbers in the DATA statement, and the numbers will have printed out.

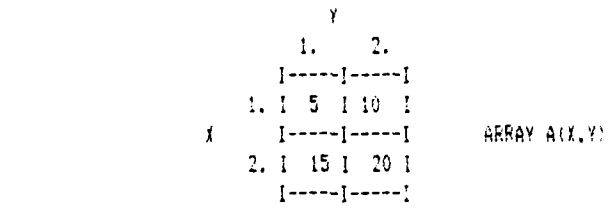
press ENTER?

```
        Y
     1.    2.
   I-----I-----I
```



Another way to look at the ARRAY is to visualize it as above. The ARRAY has four of its pockets loaded (we ignored the 0 pockets so the explanation would be simpler. They are still there, they are just not used) When A(X,Y) = 10, X must equal 1 and Y must equal 2. Do you see?

press ENTER?



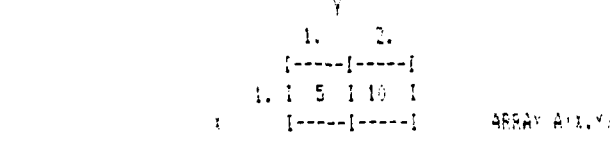
If a value of one of the pockets of array A(X,Y) equals 15, what are the values of X & Y that would reference that pocket?

A X=1, Y=1
B X=1, Y=2
C X=2, Y=1
D X=2, Y=2

ENTER the letter opposite the correct answer?

WRONG - the correct answer is C

press ENTER?



```
           2. I  15 I  20 I
              I-----I-----I
```

If X = 1 and Y = 1, what value is in the pocket pointed to by
ARRAY A(X,Y)?

A 10   B 15
C 20   D   5

ENTER the letter opposite the correct answer?

WRONG - the correct answer is D

press ENTER?

```
                 Y
              1.     2.
              I-----I-----I
           1. I  5  I 10  I
        X     I-----I-----I      ARRAY A(X,Y)
           2. I  15 I  20 I
              I-----I-----I
```

The FIRST subscript in an ARRAY ALWAYS indicates the ROWs of
DATA, and the SECOND subscript indicates the COLUMNs of data.
Therefore, ARRAY A(X,Y) has X rows and Y columns. You will
find that once you visualize a two dimensional ARRAY, the
others will come quite easily.

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

Dimension & Clear

Whenever you use an ARRAY in BASIC, the processor has to make room for all the extra memory pockets that you will use. Up to a limit, the machine can handle unexpected ARRAYs, but after you designate more than 10 pockets or 3 dimensions, the computer must know ahead of time so it can reserve enough space. The way you tell the computer to use an ARRAY with at least one subscript bigger than 10 or with more than 3 dimensions to it (ie N(3,3,3,3)), you use the DIM statement. DIM stands for DIMENSION, and it must be used before the ARRAY is used, and it cannot be changed once the program is RUNning.

press ENTER?

```
10 DIM B(12)
20 FOR X = 1 TO 12
30     READ B(X)
40 NEXT X
50 DATA 5,10,15,20,25,30,35,40,45,50,55,60
60 FOR X = 1 TO 12
70     PRINT B(X);
80 NEXT X
RUN

 5  10  15  20  25  30  35  40  45  50  55  60
```

The DIM statement told the computer that 12 pockets were needed and the READ, DATA, and PRINT statements filled the ARRAY and printed it out.

press ENTER?

Dimension and Clear

Another problem the computer has with memory allocation, is reserving enough room for STRING space. For every letter in a STRING, the BASIC processor must use a little over one memory

location. It must be able to find the letter once it stores it so it uses one location to store the letter, and another to remind it where it put the STRING in the first place!)
Unlike number variables, STRINGs can use up to 255 characters per line (numbers seldom use more than 4 - the reason is rather technical, you may wish to look up how data is handled internally by the computer in a reference book). If you are going to use more than 50 characters worth of STRING space, you must CLEAR more room for it. The clear statement actually WIPES OUT data space and reserves memory locations, so it must ALWAYS be the first statement if you are going to need it.

press ENTER?

Dimension and Clear

If you use the CLEAR statement in the middle of a program, the accumulation of data that you have stored in variables to the point that the CLEAR word was used, will be zeroed out.

```
10 X = 15
20 CLEAR
30 PRINT X
RUN
```

Do you see? The CLEAR word zeroed out X!.....press ENTER?

```
10 CLEAR 1000
20 DIM G(50), A$(100)
30 FOR X = 1 TO 100
40     READ A$(X)
.......etc.
```

The above shows the first 4 lines of a program that is going to use more than 1000 spaces of string space (that will give an average of 10 letters per pocket of A$(X)) and is going to use 50 pockets of the numeric array G. (note that you always get pocket 0 for free, it is normally not used)

Notice the position of the CLEAR and DIM statements.
press ENTER?

What will be the output of the following program?

```
10 A(10) = 5
20 CLEAR
30 PRINT A(10)
```

A 0 because the clear statement is in the wrong place.

B 5 - the clear statement only affects STRINGS.

C Nothing because A(10) will cause an ERROR

ENTER the letter opposite the correct answer?

WRONG - C is the right answer

press ENTER?

The CLEAR word is used to clear storage space for strings. but it also wipes out other data.

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

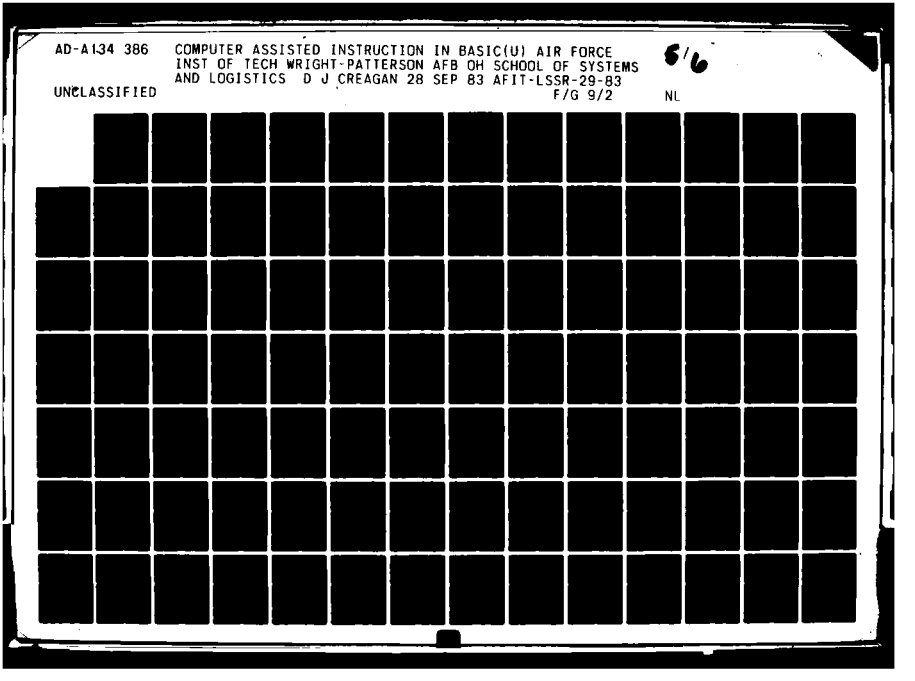
CORRECT - GoooooooooooOOOOOOOOOOOOOOOOOOD JOB!

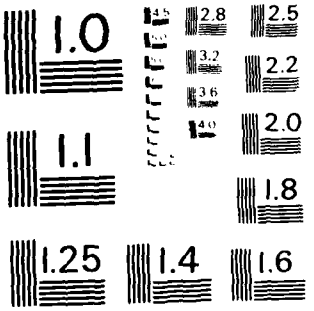
press ENTER?

As a reminder. the DIM statement does not create the ARRAY. it only defines its size. It may be used to MINIMIZE storage space by designating ARRAYs that are LESS than 10. That way

AD-A134 386 COMPUTER ASSISTED INSTRUCTION IN BASIC(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS D J CREAGAN 28 SEP 83 AFIT-LSSR-29-83

UNCLASSIFIED F/G 9/2 NL





the computer will not automatically reserve more space than needed. For example:

10 DIM A(2,2)

This would be a valid, and memory conserving statement. The processor wouldn't reserve a 10 X 10 pocket ARRAY for A(X,X), it would only reserve a 2 x 2.

press ENTER?

What is wrong with this program?

```
10 CLEAR 1000
20 DIM A(2)
30 A(2) = 5
40 PRINT A(2)
```

A The DIM statement is in the wrong place.
B The CLEAR statement is invalid.
C Nothing.

ENTER the letter opposite the correct answer?

WRONG - the correct answer is C

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

You have completed this lesson. ENTER a 'C' to Continue
to the TEST or ENTER an 'R' to start over.

ENTER your choice? C

FINAL TEST (lesson 3)

This test consists of 10 questions. you must get 70 percent
of them correct to pass. (that's 7 right out of the 10 ques-
tions). Use only capital letters in your answers. don't
include extra spaces or letters. GOOD LUCK

press ENTER to continue?

Which of the following is valid:

A KILL "SLEAZY"
B KILL SLEAZY
C UNSAVE "SLEAZY"
D UNSAVE SLEAZY

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

Does the FOR NEXT combination have to be used to set up a loop?

A YES
B NO

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?

```
Write out the first line of a FOR NEXT loop using I as the
variable. start the loop at 1 and end it at 5. use a STEP of 2.
Use line number 50.  Use all caps. leave one space between all
terms.

ENTER your answer?

WRONG - the correct answer is 50 FOR I = 1 TO 5 STEP 2
        See part 1, FOR - NEXT.

press ENTER?



10 FOR X = 1 TO 5 STEP 2
20     PRINT X;
30 NEXT X
RUN

ENTER the output from this program. leave one space between
terms.   (hint: be sure to consider the semi-colon in line 20)

ENTER your answer? 1 3 5

CORRECT

press ENTER?



How many pockets (or elements) can an array have without a
DIM statement?

A  10  not counting the 0 element
B  11  not counting the 0 element
C   3  not counting the 0 element
D   4  not counting the 0 element

ENTER the letter opposite the correct answer? A
```

CORRECT

press ENTER?

How many pockets are there in a 2 X 3 array?

ENTER your answer? 6

CORRECT

press ENTER?

How many ROWS does the following array have?

A(10,5)

ENTER your answer? 10

CORRECT

press ENTER?

Does the following array need to be DIMensioned?

A(2,3,2,2)

ENTER YES or NO? YES

CORRECT

press ENTER?

Which of the following is valid

```
A  A$(1)
B  (A$)1
C  A$(A$,2)
D  (A)(1,2$)

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?



The CLEAR statement is used for clearing STRING space and
ALWAYs must be used if your STRING use is greater than 100
but doesn't have to be used if your STRING will be LESS than
100 characters.

Is the above question TRUE or FALSE?

ENTER TRUE or FALSE? TBUE

WRONG - the correct answer is FALSE
      CLEAR statement is for clearing any space more than 50
      and it also initializes numeric variables to 0.
      See part 2, CLEAR.

press ENTER?



You have finished the test, out of 10 possible correct answers
you scored  8 .

YOU HAVE PASSED
YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:

   part 1, LOOPs
   part 2, CLEAR statement

press ENTER?
```

```
Do you want your score recorded on a permanent file?

A  YES
B  NO

Which? B

You are now qualified to go to LESSON 4.

If you want a homework assignment, select it now.

Do you want to see your homework (Y/N)?

Break in 3720
Ready
SYSTEM"RESEET*00
```

```
TRSDOS Ready
BASIC
BASIC 01.00.00 for TRSDOS Version 6
Copyright (c) 1983 by Microsoft. Licensed to Tandy Corporation.
All rights reserved.

Ready
RUN"LESSON4



LESSON:  BASIC 4            VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON:  About 1.5 hours


AUTHOR:  Capt Danny J. Cregan
         Air Force Institute of Technology

OBJECTIVE:  To teach the student how to make the computer
            communicate with standard peripheral devices.




Press the ENTER key to continue?



                    LESSON 4

This is the first part of a two part lesson.  It is divided
into the following sections:

1)  Introduction            3)  Sequential Files Intro.
2)  LPRINT & LLIST          4)  OPENing files



A  I'm taking this part in its entirety.
B  I wish to review selected areas.
C  I want to go to the second part.
```

D I want to return to the MENU.

Press either capital A, B, C, or D and then press ENTER? A

Introduction

In this lesson we will start learning how to communicate with our disk drives and printers. The first section deals with printing our programs on a printer. We can either print the output (ie the answer that our program calculated), or we can print our program listing to a printer.

The remaining sections will show us how to store data on a disk, so that we can save important calculations for future use.

Both sections are very important to the programmer and you will find yourself using them often.

press ENTER

Introduction

From this lesson to the end of your training, we will be covering areas that are complex and difficult to remember. Therefore, we recommend that you have your BASIC manual with you at all times. When we ask you a question, and you are not sure about the answer, LOOK IT UP IN YOUR MANUAL. The answer will also be in the lesson, but you should get used to using the manual. You cannot memorize all the rules in a few weeks or months. So be sure to keep your reference book handy, whether you are taking a test, or making your own program.

Remember, looking up the answers is not cheating, its LEARNING.

press ENTER

LPRINT & LLIST

LPRINT and LLIST allow you to output information to the line printer. They are extremely simple to use and they work almost exactly like the PRINT and LIST words. There are only a couple of things you need to keep in mind when you use them.

First, make sure the printer is hooked up and turned on, and, if you are using a Cromemco, make sure the printer is LINKed to your terminal. Ask your operator how to use the LINK command.

Second, make sure there is enough paper in the printer, to do your whole job.

press ENTER

LPRINT & LLIST

To LPRINT a STRING to the printer, you must enclose it in quotes, just like the PRINT statement. When you LPRINT a numerical variable, you do not enclose it in quotes. You will not see either on the screen. They will only print on the printer. Examples of valid LPRINT statements are:

```
10 LPRINT "MONTHLY TRAINING REPORT"

10 A = 10
20 LPRINT A
```

press ENTER

LPRINT & LLIST

You may LPRINT TABs also. Just as you can PRINT TABs, however, the TAB function reacts differently on different machines. The general format for LPRINTing a TAB is:

```
10 LPRINT TAB(X); data
```

where X is a number between 1 and the length of your printer carriage. When used, the carriage will go over X columns before it starts to print. Some computers will not TAB past 80 columns. You may wish to experiment with yours to see what its limitations are. Note that there isn't a space between the TAB command and the data.

press ENTER

LPRINT & LLIST

LLIST works just like LIST only it outputs to the printer

It is normally used from the IMMEDIATE mode when you want to see your program lines on paper. It is sometimes easier to find 'bugs' in your program if you can see it on a piece of paper.

LLIST, and LLIST 100-400 are valid LLIST commands. LLIST 100-400 lists lines 100-400 to the printer.

press ENTER

LPRINT and LLIST output data to the printer and to the screen.

Is the above sentence TRUE or FALSE

A TRUE
B FALSE

ENTER the letter opposite the correct answer and press ENTER ?

CORRECT

press ENTER

Which of the following commands will LLIST your entire
program to the printer?

A LLIST
B LLIST ALL
C LLIST "ALL"
D LLIST "lfn" (lfn is your program name)

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

Sequential Files Intro

Remember, in the previous lessons, we learned that we could
INPUT data into our programs, but that the data was not stored
permanently. If we turned the machine off or otherwise ended
our program, all the data that we ENTERed was lost. If we
wanted to RUN the program again, we had to re-ENTER the data.

In the rest of this lesson, we will learn how to store our data
that we ENTERed on a disk FILE. When we do that, we can always
recall it for future use, and we won't have to keep entering
the same information everytime we run a program. We just have
to tell the program to read the data from a disk. Our life
with our computer then becomes much easier.

press ENTER?

Sequential Files

A disk file is an organized collection of data, such as a training record, or a mailing list. It is usually composed of JUST the data, and nothing else. Program statements or BASIC words are normally not stored in file format.

press ENTER

Sequential Files

To transfer data from a BASIC program to a disk file, you must create a BUFFER in memory. The data is first transferred to the BUFFER, and then it is processed and written to the disk. There are two types of files that we can use in BASIC. They are SEQUENTIAL files and DIRECT access files. We will only explain SEQUENTIAL files in this lesson. If you need to learn DIRECT access files, this lesson will still help you because many of the commands are similar. Your BASIC manual will explain the differences. DIRECT access is often called "RANDOM" access by some manuals.

press ENTER

Is the following statement TRUE or FALSE?

Sequential files do not need a BUFFER in memory, but DIRECT access files do.

A - TRUE
B - FALSE
C - I DON'T KNOW

ENTER the letter opposite the correct answer? A

WRONG - You ALWAYS have to create a BUFFER in memory.

we will show you how in the next parts of the lesson.

press ENTER?

Sequential Files

With a sequential file, you must access the data the same way you wrote it to the disk. If you were to write the ages of three people to the disk, using sequential access mode, you could only read the last age you entered by reading ages one and two first. They would be stored in a line, and the computer would have to start with the first age and search through the list in order, until it found the last age. Even though this is not a fast way of handling files, the computer still does a good job of it. Your scores for the previous tests were logged into a file using SEQUENTIAL access mode.

press ENTER?

If you wrote two names to a disk file using SEQUENTIAL access mode, could you get to the LAST name you wrote without reading the FIRST name you wrote?

A YES
B NO
C I wish to take the 5th

ENTER the letter opposite the correct answer? A

WRONG - the correct answer is B

press ENTER?

Sequential Files Intro

The statements and functions used with sequential files are:

```
     OPEN            PRINT#          EOF
     CLOSE           INPUT#          CLOSE
```

We will be covering these words in greater detail in the following sections. You don't have to memorize them now, just look them over. Generally, you would OPEN the file, either INPUT# your data, or PRINT# it out to the disk, and then you would CLOSE the file before going on.

press ENTER^

```
10 OPEN "O",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1
```

This is an example of how to create a SEQUENTIAL file, write some data to it, and then CLOSE it. We will be examining this program, and a companion program that will INPUT data from the file we created, in the next sections. Generally speaking line 10 OPENs the file (we'll explain more later), line 20 asks the operator to INPUT his/her name, and line 30 writes the name out to a file called "TEST". Line 40 CLOSEs the file.

press ENTER^

Sequential access files are written to a disk in order, and you cannot access a piece of information from the middle of the file without searching through all the records (or data pieces) from first to the one that has the record you want.

Is the above statement TRUE or FALSE^

A TRUE
B FALSE
C Uhhh... somewhere in-between^

ENTER the letter opposite the correct answer^ B

```
WRONG - This is a new concept. please go back and
        review this section before going on.  You will be
        given the opportunity to review in a few moments

press ENTER?



The two types of file modes, SEQUENTIAL and DIRECT.

Is the above statement TRUE or FALSE?

A  TRUE
B  FALSE

ENTER the letter opposite the correct answer? A

CORRECT !

press ENTER?



Which do you wish to do?

A  Continue on
B  Review this section again

press the letter opposite the correct answer and press ENTER? A



                              OPEN statement

10 OPEN "D",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1

In the above program, line 10 OPENs the file we wish to save.
Whenever you work with a file, you MUST OPEN it first, then
manipulate the data, and then CLOSE it.  If you try to write
DATA to a disk without OPENing the file, you will get an ERROR
```

message. (also, don't OPEN a file that is already OPEN)

press ENTER?

Can you OPEN a file that is already OPEN?

A YES
B NO

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?

```
10 OPEN "O",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1
```

Line 10 OPENS the file in this manner:

OPEN is the keyword that signals the computer to expect three more pieces of information. The first piece is the letter 'O' or 'I'. 'O' stands for OUTPUT, and I stands for INPUT. The next bit of data is the BUFFER number. In Microsoft BASIC you may have up to 8 buffers (more on some versions). For our purposes, we will use buffer #1. The last bit of data is the filename. Notice that the file mode and filename are in quotes

press ENTER?

```
10 OPEN "O",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1
```

To recap then, the format for the OPEN statement is:

OPEN "(mode)",buffer #,"(filename)"

If you are OUTPUTing DATA the mode is 'O', if you are INPUTing data, the mode is 'I'. You can have up to 8 buffers (you must declare any buffers over 3 when working with the TRS-80 just answer the BASIC startup dialog with the correct number of files - 3 is the default). The filename must be in quotes.

press ENTER?

What are the two file modes?

A INPUT and OUTPUT
B "I" and "O"
C 1 through 4 and 4 through 8
D None of the above

ENTER the letter opposite the correct answer? B

CORRECT - Way to go!

press ENTER?

OPEN statement

```
10 OPEN "I",1,"TEST"
20 INPUT#1,N$
30 PRINT N$
40 CLOSE 1
```

Here is an example of an OPEN statement (line 10) that opens a file for INPUT. Notice that the mode is 'I'.

Also notice that there is NOT a comma between the OPEN word and the MODE, but that all the rest of the terms are separated by a comma.

press ENTER

What is the significance of file modes 'O' and 'I'?

A 'O' tells the computer that you are going to write to disk
and 'I' tells the computer you are going to input from disk

B 'O' tells the computer that the files section is ON and
'I' tells the computer that you want to INTERROGATE

C 'O' tells the computer you want to Organize files and 'I'
tells NASA to launch the shuttle.

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER

OPEN Statement

```
10 OPEN "O",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1
```

The buffer number can be any number between 1 and 3. If we use the buffer for one file, and later we OPEN another file in the same program, we cannot use the same buffer number. It MUST BE DIFFERENT. If there is more than one file OPEN at the same time, then they must be using different buffers!!

press ENTER

How many files with the same buffer number can we have OPEN at the same time?

```
***** Listing of Program "LESSON4/TXT" *****                    07-11-87 - 12:12:11


A  ONE          B  TWO          C THREE        D  FOUR

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?



10 OPEN "O",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1

You must enclose the filename in quotes if it is a character
string.  However, the following is also legal:

10 A$ = "TEST"
20 OPEN "O",1,A$

press ENTER?



10 A$ = "TEST"
20 OPEN "O",1,A$

Note that a STRING variable may take the place of the filename
AS LONG AS THE STRING VARIABLE IS SET EQUAL TO A VALID NAME !!

If you use a STRING variable for a filename, you do not enclose
it in quotes.  Look at the example above.

press ENTER?



Is the following statement valid?

10 OPEN"I",2,"A:B"
```

```
A  YES
B  NO

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?



10 OPEN "O",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1

When line 10 OPENs the file 'TEST', the computer searches
available disk space to see if the file already exists. IF IT
DOESN'T EXIST, THE COMPUTER WILL CREATE IT AUTOMATICALLY!!
(this is only true for the OUTPUT mode. if INPUT mode, the file
must have existed previously or BASIC will print an ERROR)

REMEMBER ALSO, unless you've CLOSED a file that has previously
been opened, you cannot use the buffer number again !

press ENTER?



10 OPEN "O",1,"TEST"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1

If the file 'TEST' exists, then line 10 will re-open it, line
20 will WRITE OVER THE PREVIOUS DATA IN THE FILE, and line
40 will CLOSE the file.  EVEN IF THE FILE WAS SEVERAL THOUSAND
WORDS LONG, AFTER THE ABOVE PROGRAM IS RUN, IT WILL ONLY BE
AS LONG AS THE NAME THAT WAS ENTERED IN LINE 20! !

Sequential files must be loaded into memory, manipulated and
then written back out in their entirety. You cannot just write
a single record onto the front of the file that's on the disk.
```

press ENTER?

Write in the statement that will OPEN a file for OUTPUT to disk and use buffer number 3. The filename is TWIG

Do not use a line number (although you would normally), and do NOT put in ANY blanks.

ENTER your answer ?

WRONG - this section is difficult to visualize sometimes. However, it is very important. You may wish to review it after this quiz. The correct answer is: OPEN"O",3,"TWIG"

press ENTER?

When a sequential file is OPENed that formerly had your training records in it, and you didn't want the DATA destroyed, what must you be sure to do?

A Not write in the middle of the file unless it is mode 'O'
B INPUT# all the data, manipulate it, then PRINT# it back out
C Nothing, you cannot OPEN a file that was previously created
D Write only on the END of the file

ENTER the letter opposite the correct answer?

WRONG - the correct answer is B

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

You have completed this portion of lesson 4. If you wish
to continue on to the next half of the lesson, enter a 'C'.
If you wish to review this lesson again, enter an 'R'.

Enter your choice now?

You have completed this portion of lesson 4. If you wish
to continue on to the next half of the lesson, enter a 'C'.
If you wish to review this lesson again, enter an 'R'.

Enter your choice now?

You have completed this portion of lesson 4. If you wish
to continue on to the next half of the lesson, enter a 'C'.
If you wish to review this lesson again, enter an 'R'.

Enter your choice now?

You have completed this portion of lesson 4. If you wish
to continue on to the next half of the lesson, enter a 'C'.
If you wish to review this lesson again, enter an 'R'.

Enter your choice now?

You have completed this portion of lesson 4. If you wish
to continue on to the next half of the lesson, enter a 'C'.
If you wish to review this lesson again, enter an 'R'.

Enter your choice now? C

Going to the second half of the lesson. wait one moment

LESSON 4

This is the second part of a two part lesson. It is divided into the following sections:

1) CLOSE
2) PRINT#
3) INPUT$ & EOF
4) SUMMARY
5) TEST

A I'm taking this part in its entirety.
B I wish to review selected areas (or take the test).
C I want to return to the Menu.
D I want to go the the first part.

Press either capital A, B, C or D and then press ENTER? A

CLOSE Statement

We already stated previously that the CLOSE statement was necessary after you were done manipulating your files. It has a few variations that are nice to know.

CLOSE [buffer #,, ...]

The format for CLOSE is the BASIC word CLOSE plus an optional buffer number. If you include a buffer number, just that buffer will be closed. If you do not include a buffer number then ALL the buffers that were OPEN will be closed!

press ENTER?

Is the following program valid (lines 20 & 30 are good:

```
10  OPEN"O",3,"QUESTION"
20  INPUT A$
30  PRINT#3,A$
40  CLOSE
50  OPEN"O",3,"QUESTION"
.....etc
```

A YES it is valid
B NO it will fail because line 50 OPENs a file already used

ENTER the letter opposite the correct answer?

WRONG - file #3 was CLOSEd by line 40, so buffer 3 could be
used again in line 50.

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

PRINT #

```
10 OPEN "O",1,"TES"
20 INPUT"Type in your name and press ENTER";T$
30 PRINT#1,T$
40 CLOSE 1
```

In line 30 we PRINT to the file buffer (that we OPENed in line
10) the variable T$. Notice that the number to the right of
the PRINT# corresponds to the buffer number. If the buffer
was number 3, then, after we OPENed the file, we would
PRINT#3,T$ in line 30

press ENTER?

PRINT #

PRINT# prints items to a sequential disk file. When you first OPEN the file, the computer sets a pointer at the beginning of the file. When you tell the computer to PRINT# something, it starts writing data to the disk at the place where the pointer is. At the end of the PRINT# operation, the pointer advances, so values are written in sequence. PRINT# writes data to the disk almost exactly the way PRINT writes data to the screen (or LPRINT writes to the printer).

press ENTER?

PRINT#

Commas and semi-colons react the same way with PRINT# that they do with PRINT statements. If you were to write the following program:

```
10  A=10.3
20  B= 20.2
```

to a disk using PRINT#1,A,B (as opposed to PRINT#1,A;B), then you would put this on the disk : 10.3   20.2

See the extra spaces? Those are 13 blanks that BASIC writes to the disk.

press ENTER?

PRINT#1,A,B

The above command puts this on disk : 10.3   20.2

If you use a semicolon, like this, PRINT#1,A;B then you get:

 10.3 20.2

There are only three spaces between the numbers. So to save space on the disk, you may want to write to disk using semicolons instead of commas between your variables. Either way will work, it's just that the commas cause 10 extra blanks to be PRINT# 'd to the disk.

press ENTER?

What is wrong with the following program?

```
10 OPEN"I",1,"NEWPROG"
20 PRINT#1,34;22;55
30 CLOSE 1
```

A The mode is incorrect
B The file buffer is incorrect
C You cannot use semicolons between numbers in a PRINT#
D Nothing

Enter the letter opposite the correct answer? B

WRONG - The correct answer is A - the mode is incorrect for
PRINT#ing.

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

INPUT# & EOF

INPUT# is similar to the INPUT word that we learned earlier only it INPUTs data from a disk that previously had data

printed to it.

```
10 OPEN "I",1,"TEST"
20 INPUT#1,N$
30 PRINT N$
40 CLOSE 1
```

The format for the INPUT# statement is similar to the PRINT# statement. It is: INPUT# (buffer#),(variable1),(var2),(etc)

press ENTER?

INPUT#

INPUT# inputs data from a sequential disk file and stores the data in a variable. INPUT# doesn't care how data was placed on the disk. It could have been put there with one PRINT# or twenty PRINT# statements. WHAT MATTERS TO INPUT# IS HOW THE DATA IS TERMINATED ON THE DISK, AND WHAT KIND OF DATA IT IS INPUTTING.

press ENTER?

Does the INPUT# statement check to see how the data was placed on the disk, or does it check to see how the data is terminated?

A It only checks to see how it was terminated - it doesn't care how the data got there.

B It checks to see how the data was placed. It makes a difference how many PRINT# statements were used.

C Now THIS question is easy. Just give me a second and I'll think of the answer. Hmmm. let's see....No, don't tell me..

ENTER the letter opposite the correct answer? A

CORRECT - good job

press ENTER?

INPUT# & EOF

If we are inputting STRING data (our variable is a STRING such as INPUT#1,N$), INPUT# starts putting data into the variable starting with the first NON-SPACE it encounters in the file, and ending when it encounters a carriage return or a comma, or EOF marker (more about EOF later).

If the variable is numeric, then INPUT# fills the variable with the first character that is not a space or carriage return, and stops when it encounters another space, comma, or carriage return or EOF marker.

press ENTER?

INPUT# and EOF

Here is an important concept to understand about how INPUT# works when you use STRINGs. IF YOU PRINT A STRING TO DISK, AND YOU PRINT A NUMBER WITH IT, IT WILL LOOK LIKE THIS ON DISK:

STRING DATA HERE WITH NUMBER FOLLOWING 1001

IF YOU INPUT THIS DATA, YOU MUST INPUT IT USING A STRING VARIABLE. WHEN YOU DO, THE COMPUTER WILL PACK THE STRING WITH ALL THE DATA ON THE LINE, (INCLUDING THE NUMBER!) BECAUSE A STRING INPUT DOES NOT RECOGNIZE SPACES AS TERMINATORS. IT ONLY RECOGNIZES CARRIAGE RETURNS AND COMMAS

Why is this important? Because you wrote the data to disk with TWO variables, and you read it back with only ONE. If you tried to read the number after you read in the string, you would not find it!

press ENTER?

The solution to the problem of how to write STRING data and NUMERIC data to disk, is to separate them with carriage returns (or ENTERs). That way you can read them back with INPUT# statement in the same way you wrote it. For example, if you write a string and a number to disk, do it this way:

```
10 ..... program assumes file opened correctly.....
20 PRINT#1,"PRINT THE STRING WITH ONE LINE and the number with another."
30 PRINT#1, 1001
```

And when you read it back, use two separate statements.

```
10 INPUT#1,A$
20 INPUT#1,N
```

This will solve the problem of mixing strings and numbers on disk.

press ENTER?

```
10 OPEN "I",1,"TEST"
20 INPUT#1,N$
30 PRINT N$
40 CLOSE 1
```

If we were to run this program, and the name that was in the first record in the file was 'DANNY JOE', line 20 would start with the first byte of the first record that was not a space or a carriage return, and load N$ with it. It would continue loading N$ until it encountered either a carriage return or a comma. If the data contained a quoted character string, then all the data between the quotes would be stuffed into N$ (unless a comma or carriage return were encountered).

press ENTER?

```
10 OPEN "I",1,"TEST"
20 INPUT#1,N$
30 PRINT N$
40 CLOSE 1
```

Eventually, and very quickly, the string would be loaded with the characters 'DANNY JOE', and line 30 would print them out.

But what would happen if there was nothing in the file? The INPUT$ statement in line 20 would encounter the EOF marker and an ERROR would be returned saying that the computer tried to input data that wasn't there.

There is a way to test to see if the file is at the end or if it is empty. *Press ENTER to see what it is?*

The very first thing a computer does when it OPENs a file, is it puts a marker on the end of it. The marker is called an EOF marker. If we wanted to find out if the end of a file had been reached, or if the file was empty, we would test it like this:

```
10 OPEN"I",1,"TEST"
20 IF EOF(1) THEN GOTO 60
30 INPUT#1,N$
40 PRINT N$
50 GOTO 20
60 PRINT"END OF FILE ENCOUNTERED"
70 CLOSE 1
.....more of the program or an END statement
```

press ENTER?

```
10 OPEN"I",1,"TEST"
20 IF EOF(1) THEN GOTO 60
30 INPUT#1,N$
40 PRINT N$
50 GOTO 20
60 PRINT"END OF FILE ENCOUNTERED"
70 CLOSE 1
.....more of the program or an END statement
```

If we had printed a thousand and one names into the file 'TEST' the computer would put the EOF marker in place # 1002, and when we used the EOF buffer #, test after name # 1001, control

would pass to line 30. All the names would have been printed"
Notice that we never re-OPEN a file if we haven't CLOSED it.
If we tried to re-OPEN an already OPEN file, we'ld get an ERROR

press ENTER?

INPUT# and EOF

The format for the EOF statement is :

EOF(buffer #)

where buffer number corresponds to the buffer number of the
file you are testing .

Remember to only use the EOF test on a file that is OPEN !

press ENTER?

Suppose we have a file called "TEST" that is full of numeric
data. What is wrong with the following program if we were
trying to print the file out to the screen"

```
10 OPEN"I",1,"TEST"
20 IF EOF(1) THEN STOP
30 INPUT#1,N
40 PRINT N
50 GOTO 10
```

A Nothing
B If "TEST" is empty, the EOF check won't catch it
C Line 50 should be GOTO 20
D The file mode is incorrect

ENTER the letter opposite the correct answer?

WRONG - The problem is that the program trys to re-OPEN the
        file that hasn't been CLOSEd. Correct answer is C

press ENTER?

What is wrong with this program?

```
10 OPEN"I",1,"TEST"
20 IF EOF(2) THEN STOP
30 INPUT#1,N$
40 PRINT N$
50 GOTO 20
```

A The wrong buffer number is used
B Nothing
C There is no END statement

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

SUMMARY

In this lesson we have learned a great deal about file input and output. However, there is a great deal of information that we have not covered.

The purpose of this lesson was to introduce you to the fundamental ideas behind sequential files. You should combine this knowledge with the previous lessons, and do some outside studying on your own. After you take the test, you will be given an assignment that will include many of the techniques

we have already learned.

On the following pages, there is a program that uses what we have learned in this lesson. Study it carefully.
press ENTER?

The purpose of the program on the following page is to update a data file that contains STRING data. It reads in a file and simultaneously writes out the same data to a different file. When you update a file this way, you end up with an updated file that has a different name than the one you started with. While you are studying the program, think about how you would do it without changing the filename.

HINT: the program would have to read the data into an array and then add data onto the array, and finally write the whole array back out to the old file.

Note how the comments are inserted. Look up why this is legal in your BASIC manual. You may be surprised!

press ENTER?

```
10 CLEAR 500              :'Clears string space
20 INPUT"Input filename";I$
30 INPUT"Output filename";O$
40 OPEN"I",1,I$           :'OPENs the INPUT file
50 OPEN"O",2,O$           :'OPENs the OUTPUT file
60 IF EOF(1) GOTO 100     :'Checks for end of file in file #1
70 INPUT#1,D$             :'INPUTs data from file #1
80 PRINT#2,D$             :'PRINTs data to the new file
90 GOTO 60                :'Goes back for more from file #1
100 INPUT"ENTER DATA  enter 999 to stop";D$
110 IF D$ = "999" GOTO 140
120 PRINT#2,D$            :'Updates new file with your data
130 GOTO 100              :'Goes back until line 110 sees 999
140 CLOSE
```

We assume a legal input file previously existed...press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

You are now done with this lesson. If you wish to continue
to the test, ENTER a T. If you want to review, ENTER an R.
ENTER your choice?

You are now done with this lesson. If you wish to continue
to the test, ENTER a T. If you want to review, ENTER an R.
ENTER your choice? T

Going to TEST 4 - please standby

FINAL TEST (lesson 4)

This test consists of 10 questions. You must get 70 percent
of them correct to pass. (that's 7 right out of the 10 ques-
tions!). Use only capital letters in your answers, don't
include extra spaces or letters. When you successfully complete
the test, you will be given a homework assignment that will
bring many things together for you. GOOD LUCK!

press ENTER to continue?

When you LPRINT data you must be sure the printer is on, has enough paper, and is properly connected. LPRINTing does not send output to the screen.

Is the above paragraph TRUE or FALSE?

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

What is the command to list your program to the printer?

A LPRINT "(filename)"
B LIST "(filename)"
C LLIST
D None of the above

ENTER the letter opposite the correct answer? C

CORRECT

press ENTER?

You may access the middle record of a sequential file without reading in the records that are in front of it.

Is the above statement TRUE or FALSE

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

WRONG - the correct answer is B
The only way to access a record in the middle of the

file is to read the records in front of it.
See part 1. Sequential Files.

press ENTER?

To transfer data from your file to a disk, the computer must first send the data to a buffer where it is processed. the same is true for transferring data from the disk back to the computer.

Is the above paragraph TRUE or FALSE?

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

You wish to access a previously created disk file. You will read in the data and use it to make an important financial decision. Which of the following statements will open the file and read the data into buffer 5. The filename is B52DAT

A 10 OPEN"O",1,"B52DAT"
B 20 OPEN"5",1,"B52DAT"
C 30 OPEN"I",5,B52DAT
D 40 OPEN"I,5,","B52DAT"
E None of the above

ENTER the letter opposite the correct answer? E

CORRECT

press ENTER?

What is wrong with the program below (assume the END statement in line 20 automatically closes the files when it is executed)

```
10 OPEN"I",1,"TEST"
20 IF EOF(1) THEN END
30 INPUT#1,A$
40 PRINT A$
50 GOTO 10
```

A Nothing
B A 'File already open' ERROR will be generated
C The wrong file mode is used
D Line 20 is illegal

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?

The CLOSE word may be used without a buffer number; however, when you do that, only the most recent file is CLOSEd.

Is the above statement TRUE or FALSE?

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

What is wrong with the following program?

```
10  OPEN"O",1,"TEST"
20      PRINT#1,10;20;30
```

```
30 CLOSE


A Nothing
B Wrong file mode is used
C The CLOSE statement is invalid
D The PRINT#1 statement should be INPUT#1

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?



You have opened a file correctly, and you now want to read in
the data from it.  Type in the command you would use.
Use line number 10, a space, and the command.  Use buffer #3
and read  the data into variable A$

ENTER the command now ? 10 INPUT#3,A$

CORRECT

press ENTER?



What is wrong with the following program?

10 IF EOF(1) THEN GOTO 40
20 INPUT#1,A$
30 GOTO 10
40 CLOSE
50 END

A Nothing
B The file wasn't properly opened
C Nothing will happen with the data
D The END statement is not needed

ENTER the letter opposite the correct answer? B
```

```
***** Listing of Program 'LESSON4/TXT' *****                    07/11/80 - 02:10:01

CORRECT

press ENTER?




You have finished the test. out of 10 possible correct answers
you scored  9 .

YOU HAVE PASSED
YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:

     Sequential Files

press ENTER?




Do you want your score recorded on a permanent file?

A  YES
B  NO

which? B




You are now qualified to go to LESSON 5.


Do you wish a homework assignment (Y/N)?

Break in 4100
Ready
SYSTEM"RESET #00
```

```
TRSDOS Ready
basic
BASIC v1.00.00 for TRSDOS Version 6
Copyright (c) 1980 By Microsoft, Licensed to Tandy Corporation.
All rights reserved.

Ready.
run"lesson5"



LESSON:  BASIC 5            VERSION: 1 AUGUST 80

TIME REQUIRED TO COMPLETE LESSON:  About one hour


AUTHOR:  Capt Barry J. Dreegan
         Air Force Institute of Technology

OBJECTIVE:  To teach the student how to use SUBROUTINES
            and LIBRARY functions.




press the ENTER key to continue


                    LESSON 5

This is the first part of a two part lesson
It is divided into the following sections.

1   Introduction          4   Nested Subroutines
2   Subroutines           5)  ON GOSUB
3   GOSUB & RETURN


a  I'm taking this part in its entirety.
b  I want to review selected areas.
c  I want to go to the second part.
d  I want to return to the menu.
```

Press either capital A, B, C, or D and then press ENTER: A

INTRODUCTION

In the last episode of our computerized book, we assigned you a homework problem that dealt with disk I/O (disk input/output). It included most of the concepts that we have been studying. Now we are on the downhill part of the course. That's right, you are almost done with the hard parts of BASIC.

The last obstacle is learning about SUBROUTINES, which you will learn in the first half of this lesson. After that, we will review the numerous library functions (but not have to memorize them, we'll just have to know how they work), and then go on to lesson 6.

Press ENTER

INTRODUCTION

Somewhere during each lesson, we emphasize the value of doing extra study besides what this course teaches you. We all learn more when we DO something that we have read about. This is certainly true with learning a programming language.

For now, though, get out your favorite BASIC manual, and curl up beside your computer for another lesson in BASIC!

Press ENTER

Which do you wish to do?

A Continue or
B Review this section again

Press the letter opposite the correct answer and press ENTER: A

SUBROUTINES

We use the word "ROUTINE" to describe the statements in the body of a program. "SUBROUTINE" is used to describe a miniprogram that was built, attached to the main program, and used to perform a mini-task that, for some reason, needs special attention.

Usually, we use subroutines to do tasks which we perform often in our program such as printing out a menu after each module of a CAI program is finished by a student. That way, we only have to write the menu program once and call on it when we need it.

press ENTER

Why would we want to use a subroutine?

[illegible] To do those parts of the program that are used often
B To go UNDER the normal routines

ENTER the letter opposite the correct answer?

WRONG - this was supposed to be an easy question to answer. Obviously we screwed up somehow. We will send you back to the beginning of this part. Try reading between the lines a little.

press ENTER

SUBROUTINES

We use the word "ROUTINE" to describe the statements in the body of a program. "SUBROUTINE" is used to describe a miniprogram that was built, attached to the main program, and used to perform a mini-task that, for some reason, needs special attention.

Usually, we use subroutines to do tasks which we perform often in our program such as printing out a menu after each module of a CAI program is finished by a student. That way, we only have to write the menu program once and call on it when we need it.

Press ENTER

Why would we want to use a subroutine?

A To do those parts of the program that are used often
B To go UNDER the normal routines

ENTER the letter opposite the correct answer? A

Correct - Good Job!

Press ENTER

SUBROUTINES

Subroutines are mini-programs that we put in our main program and, when we need them, we can go to them, perform the task they are built to do, and return to the main program.

Subroutines differ from other forms of program control in that they ALWAYS RETURN CONTROL TO THE STATEMENT THAT FOLLOWS THE STATEMENT THAT CALLED THEM. That means you can call a subroutine anywhere within a program, and the computer will remember where the call came from, and when the subroutine has done its job, control will go back to the statement that followed the calling statement.

Press ENTER

SUBROUTINES

If you have built a program that has to continually print out a prompt, asking the users if they want to review previous sections of the program that have run, you would likely use a subroutine to ask the question, return to the main program with the answer stored in a variable, and branch to the right part of the program, based on the answer.

The following is an example such a program

press ENTER

```
10 PRINT"A CAI program is a computer assisted instruction"
20 PRINT"program that teaches students."
30 GOSUB 10000     :'<---- LOOK use sub. after every major part
50 IF T$ = "B" THEN GOTO 10 :' T$ returns from sub with choice
60 .....ETC
80 GOSUB 10000     :'<---- LOOK use sub. after every major part
90 END   : 'You must ensure program doesn't goto sub by accident
10000 REM
10001 REM the subroutine follows
10002 REM
10003 INPUT"Do you want to continue or review (Enter B=Back)";T$
10010 RETURN
```

We will explore the main points of this routine in the next section. Press ENTER to go on

Where does control transfer when a subroutine returns to the main program?

A The beginning of the program
B The calling statement
C The statement after the calling statement
D None of the above

Enter the letter opposite the correct answer? C

CORRECT - Super job . . . Now we can go on

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

GOSUB & RETURN

The set of statements that you use to implement a subroutine is made of the GOSUB and RETURN words. The GOSUB word is used almost exactly like the GOTO statement. You put the line number or the start of the subroutine on the right of the GOSUB word. When the computer gets to it, it transfers control to the subroutine. When the subroutine is done, it returns to the main program by using the RETURN statement. You cannot GOTO the main program from a subroutine without risking disaster. You should always use the RETURN statement.

press ENTER?

```
10 print"A CAI program is a computer assisted instruction"
20 print"program that teaches students."
30 GOSUB 10000     :' ---- LOOK use sub. after every major part
50 IF T$ = "B" THEN GOTO 10 :' T$ returns from sub with choice
60 .....ETC
9, GOSUB 10000     :' ---- LOOK use sub. after every major part
9: END   :'You must ensure program doesn't goto sub by accident
10: REM
11: REM the subroutine follows
12: REM
```

```
10000 INPUT"Do you want to continue or review (A=Con B=Rev)";T$
10010 RETURN
```

Line 30 and line 80 call the subroutine, and line 10010 RETURNs to the appropriate statement.   press ENTER?

```
10 print"A CAI program is a computer assisted instruction"
20 print"program that teaches students."
30 GOSUB 10000      :'<---- LOOK use sub. after every major part
50 IF T$ = "B" THEN GOTO 10 :' T$ returns from sub with choice
60 .....ETC
80 GOSUB 10000      :'<---- LOOK use sub. after every major part
90 END   :'You must ensure program doesn't goto sub by accident
100 REM
110 REM the subroutine follows
120 REM
10000 INPUT"Do you want to continue or review (A=Con B=Rev)";T$
10010 RETURN
```

If line 30 calls the subroutine, what line gets control after the RETURN statement .... ENTER the correct line number? 50

CORRECT - You are getting GOOOOOOOooood!

press ENTER?

GOSUB & RETURN

You can have more than one GOSUB in a program, and you can have more than one RETURN in a subroutine. If you have more than one RETURN, then the computer will return when it reaches the first RETURN statement it comes to. Generally speaking, you should try to limit the number of exits from a subroutine because it can get very confusing if you have RETURNs stuck all over the place. It is usually possible to have only one exit to any program or subroutine.

press ENTER?

```
Is the following program valid?

20 INPUT"Enter a number between 0 and 20";N
30 GOSUB 100
40 END
100 IF N < 10 THEN PRINT "Number is less than 10" ELSE GOTO 120
110 RETURN
120 IF N > 10 THEN PRINT "Number is more than 10" ELSE GOTO 140
130 RETURN
140 PRINT "Your number is 10"
150 RETURN

A  Yes, but confused by too many RETURNs
B  No, the program will never get passed line 110

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?



ENTER the command to go to a subroutine that starts on line
1000.  Use line 10y and put one space between all terms.

Enter the command now? 100 GOSUB 1000

CORRECT

press ENTER?



Which do you wish to do?

A  Continue on
B  Review this section again

press the letter opposite the correct answer and press ENTER? A
```

NESTED SUBROUTINES

What do you think the output of the following is?

```
10 PRINT"Main Program"
20 GOSUB 100
30 GOSUB 200
40 END
100 PRINT"Subroutine One"
110 GOSUB 200
120 RETURN
200 PRINT"Subroutine Two"
210 RETURN
```

press ENTER for the answer?

The answer is :

```
10 PRINT"Main Program"
20 GOSUB 100
30 GOSUB 200
40 END
100 PRINT"Subroutine One"
110 GOSUB 200
120 RETURN
200 PRINT"Subroutine Two"
210 RETURN
```

```
Main Program
Subroutine One
Subroutine Two
Subroutine Two
```

press ENTER?

```
10 PRINT"Main Program"
20 GOSUB 100
30 GOSUB 200
40 END
100 PRINT"Subroutine One"
```

```
110 GOSUB 200
120 RETURN
200 PRINT"Subroutine Two"
210 RETURN
```

This is an example of a 'NESTED SUBROUTINE'. The subroutine
in line 100 calls the subroutine in line 200. It is perfectly
legal, and sometimes very valuable.

press ENTER?

What is the order of execution of the following program?

```
10 GOSUB 100
20 GOSUB 200
30 END
100 RETURN
200 GOSUB 100
210 RETURN
```

Type out the line numbers as they would be executed, putting a
space between each number. (ie 10 20 30 40)
Enter your answer?

WRONG - the correct answer is 10 100 20 200 100 210 30
That was a rough one, you may want to review this part

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

Remember the ON GOTO statement from our previous lessons?

There's a command very similar to ON GOTO that can be used with subroutines. You have probably already guessed that the command is called ON GOSUB.
Here is an example:

```
10 INPUT"ENTER A NUMBER BETWEEN 1 AND 4";N
20 ON N GOSUB 300,400,500,600
```

If N = 1 then subroutine 300 would be executed, if N = 2, subroutine 400 would execute, and so on.

PRESS ENTER?

ON GOSUB

```
10 INPUT"ENTER A NUMBER BETWEEN 1 AND 4";N
20 ON N GOSUB 300,400,500,600
```

If the value of N exceeds the number of options that are available, then ON GOSUB will default to the first available line number (in this case it would be 300)

PRESS ENTER FOR AN EXAMPLE?

```
10 PRINT"TYPE 1 FOR SQUARE TABLE"
20 PRINT"TYPE 2 FOR THE CUBE TABLE"
30 INPUT A
40 ON A GOSUB 1000,2000
50 GOTO 10
1000 FOR X = 1 TO 50
1010   PRINT X,X*X
1020 NEXT X
1030 RETURN
2000 FOR X = 1 TO 50
2010   PRINT X, X*(*X
2020 NEXT X
```

```
2030 RETURN

If you ENTER a 1, then subroutine 1000 is used.  Which subrou-
tine is used if you ENTER a 4 (1000,2000 or NONE)? NONE

CORRECT

PRESS ENTER?



Which do you wish to do?

A  Continue on
B  Review this section again

press the letter opposite the correct answer and press ENTER? A



You are now done with this part of the lesson. If you ENTER a
 ' C ' you will go on to the second part.  ENTER an ' R ' to
start over.           Which do you want (C or R)? C



                LESSON 5A

This is the second part of a two part lesson
It is divided into the following sections.

1)  Introduction          4)  User Defined Functions
2)  Functions Overview    5)  DEF Statements
3)  Library Functions     6)  TEST


A  I'm taking this part in its entirety.
B  I wish to review selected areas (or take the test).
C  I want to go to the first part.
D  I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A
```

INTRODUCTION

In this lesson we will tackle the different catagories and kinds of FUNCTIONs that are available to us in Microsoft BASIC. We will use principles already covered in previous sessions. If you have trouble with some of the concepts, then you will have to review the appropriate lesson to catch up.

However, we will not be going into great depth with our explanations. Many FUNCTIONs have very specific uses and it would not be useful for you to memorize them. The idea you should get from this part is that there are many functions available, and that when you need them, you should get out your manual and look up the specific implementation of each one.

press ENTER

Functions Overview

Remember in the second part of Lesson 1 when we described FUNCTIONs for the first time? We said that we would come back to them in another lesson. Well, this is it!

From that lesson we should remember that functions are pre-written instructions that perform commonly used operations. You can look at functions like they were mini-subroutines. Only you don't use GOSUB or RETURN statements to call them! Instead, you just use the keyword associated with the FUNCTION and the computer performs the appropriate operation automatically. In the next sections we will study two types of functions, Library, and User Functions.

press ENTER

Functions Overview

Library functions contain useful operations that have been

written and stored in the computer, and are there whenever you need them. User functions are functions that you make up by inserting the instruction to make them in your program. Then, when your program needs the special USER FUNCTION, it can call on it with a special word.

There are many functions, and, depending on the specific implementation of Microsoft BASIC, you probably have at least 12 Library functions stored in your computer. On the next screen is a list of the typical set of Library functions.

press ENTER?

Typical Library Functions

1) ABS(exp) - gives absolute value 7) LOG(exp) - gives LOG(e)
2) ATN(exp) - arctangent in radians 8) RND(0) - random numbers
3) COS(exp) - returns cosine of exp 9) SGN(exp) - sign of exp
4) EXP(exp) - natural exponential 10) SIN(exp) - sine of exp
5) FIX(exp) - gives integer of exp 11) SQR(exp) - square root
6) INT(exp) - gives integer of exp 12) TAN(exp) - tangent(exp)

exp = any appropriate numeric expression - remember, you can't use ANY number for some of the functions (SQR(-1) bombs!)

Specific examples of these functions will be given later, or you may look them up in your BASIC manual.
press ENTER?

Is the following statement TRUE or FALSE?

Library functions are predefined instructions that are stored inside the computer, and may be used at any time in your program.

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

Is the following sentence TRUE or FALSE?

User Functions are made by the user and are not normally stored
in the computer.

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

Library Functions

1) ABS(exp) - gives absolute value 7) LOG(exp) - gives LOG(e)
2) ATN(exp) - arctangent in radians 8) RND(0) - random numbers
3) COS(exp) - returns cosine of exp 9) SGN(exp) - sign of exp
4) EXP(exp) - natural exponential   10) SIN(exp) - sine of exp
5) FIX(exp) - gives integer of exp 11) SQR(exp) - square root
6) INT(exp) - gives integer of exp 12) TAN(exp) - tangent(exp)

exp = any appropriate numeric expression - remember, you
can't use ANY number for some of the functions (SQR(-1) bombs!)

Here are twelve of the most used library functions. As an

example of how much time they can save. think of the number of statements you would have to write to calculate the logarithm of a number.
press ENTER?

Library Functions

All you have to do to calculate the log of a number, say the number is 10, is to type in the statement - PRINT LOG(10)'
The example of the logarithm of 10 is exactly how we implement the library functions. Here is another example:

```
10 INPUT"Type in the number you want the square root of";N
20 PRINT SQR(N)
30 GOTO 10
```

In this example, any positive number will have its square root printed out. Any negative number will cause an error. You can't take the square root of a negative number.

press ENTER?

Library Functions

```
10 INPUT"Type in the number you want the square root of";N
20 PRINT SQR(N)
30 GOTO 10
```

Notice that the function allows you to use a variable as an argument for the number you want the square root of. In addition to this, you can use a FUNCTION as an argument for a FUNCTION'

press ENTER for an example?

Library Functions

```
10 INPUT"Enter the number you want the square root of";N
20 PRINT SQR(ABS(N))
30 GOTO 10
```

In this example, no matter what the sign of the number you enter, you will get a valid square root, because ABS(N) will give the absolute value of N (N as a positive number), and then SQR(ABS(N)) will give the square root.

You can use just about any valid numeric expression for the (exp) part of any Library Function.

press ENTER?

Is the following statement valid?

```
10 PRINT LOG(0)
```

HINT: If you are not sure, try it on a calculator

A Yes, it is valid
B No, 0 is not a valid number for this function

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?

Library Functions

```
10 INPUT"Enter the number you want the square root of";N
20 PRINT SQR(ABS(N))
30 GOTO 10
```

The expression ABS(N) is evaluated first, then the outside expression - SQR(---) - is evaluated next. Remember from the first lesson when we said that the parenthesis is the highest priority arithmetic expression? That means that any expression that is within parenthesis will be evaluated first. If more

than one set of parenthesis is used, then the expression within the inner-most set of parenthesis is evaluated first.

press ENTER?

Is the following sentence TRUE or FALSE?

You can write your own library functions in special cases.

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

WRONG - library functions are permanently stored in the computer and cannot be created, the correct answer is B

press ENTER?

In the next few screens, we will ask you questions concerning library functions. You should get out your BASIC manual and look up the answer to the questions before you attempt to enter your response.

You should give your answer in the format, FUNCTION(number). You will always be given the number for the function, and you do not have to include a line number. For example, if we ask for the function that gives the square root of 10, you would type in: SQR(10).

press ENTER?

What is the function that gives the natural log of 8? LOG(8)

CORRECT

press ENTER?

```
Name the function that gives the absolute value of -3? ABS(-3)

CORRECT

press ENTER?


What is the function that gives the arctangent of A*3?

WRONG - the correct answer is ATN(A*3)

press ENTER?


What is the function that gives the sine of .5? SIN(.5)

CORRECT

press ENTER?


Name the function to give a random number between 0 & 1? RND

WRONG - the correct answer is RND(0)

press ENTER?


Which do you wish to do?

A  Continue on
B  Review this section again

press the letter opposite the correct answer and press ENTER? A
```

User Functions

User defined functions let you make up your own functions when you can't find a library function that will do the job.

The statement you use to DEFine a user function, so the computer will know what you are doing, is called the DEF statement. The user function only applies to the program in which it was defined. When the program is abandoned, the function is no longer valid. An example of a user defined function is coming up .... but first, something a little different.

press ENTER?

Can user functions be carried over from one program to another?

A Yes
B No, they always have to be redefined
C Both A & B above.

Enter the letter opposite the correct answer? B

CORRECT

press ENTER?

User Functions

```
10 DEF FNR2 = INT(RND(0) * 10)
20 DEF FNW(A,B) = A * B/2 + (A - B)
30 X = FNR2
40 PRINT X
50 Y = FNW(X,4)
60 PRINT Y
70 END
```

There are two user functions in this program. They are defined in lines 10 and 20, and called on in lines 30, and 50. We will cover this program in greater detail in the next section.

press ENTER

Which do you wish to do?

4 Continue on
5 Review this section again

press the letter opposite the correct answer and press ENTER? 4

DEF Statement

The DEF statement is used to define a user function and it has the following format:

(line #)DEF FN(func. name)((vars)) = (exp)

You must never use the DEF statement in the IMMEDIATE mode. You always have to have a line number. In CPM and Cromemco systems, you must separate the terms DEF and FN by one space. In TRS-80s you don't have to. The (func. name) is any valid variable name. (vars) is an optional parameter that may be passed to the function. If it is used in the DEF statement, it MUST be used when it is called on. (more on that later). The term (exp) is the calculation that you wish the function to do.

press ENTER

```
10 DEF FNR1 = INT(RND(0) * 10)
20 DEF FNW(A,B) = A * B/2 + (A - B)
30 Y = FNR1
40 PRINT Y
50 Y = FNW(3,4)
60 PRINT
```

```
70 END
```

Get out your BASIC manual and look up DEF. It will show you examples similar to this. Line 10 defines a function named R2 that will be set equal to the expression on the right side of the statement. - it will return a random number between 1 and 10. Note that this user function uses a library function as part of its definition. This is legal.

press ENTER?

Can you use a library function as part of the definition of a USER function?

A YES
B NO

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

```
10 DEF FNR2 = INT(RND(0) * 10)
20 DEF FNW(A,B) = A * B/2 + (A - B)
30 X = FNR2
40 PRINT X
50 Y = FNW(X,4
60 PRINT Y
70 END
```

Line 30 calls on the function defined in line 10. When it does, X is set equal to a random number between 1 and 10 and then it is printed out in line 40. Line 50 calls on the function in line 20, but it sends two values to the DEF statement. It sends X (a random number) and the number 4. ANY valid variable can be passed to a function, even a string.

press ENTER?

```
10 DEF FNR2 = INT(RND(0) * 10)
20 DEF FNW(A,B) = A * B/2 + (A - B)
30 X = FNR2
40 PRINT X
50 Y = FNW(X,4)
60 PRINT Y
70 END
```

A string could be passed in a function, but it would have to be operated on legally within the DEF statement. In this case the DEF statement in line 20 expects two numerical variables. Note that the two variables passed in line 50 do not match the DEFined variables. This is also legal. You can view the DEF statement as its own little program. It does not know the value of any of the variables in the outside program.
press ENTER?

```
10 DEF FNR2 = INT(RND(0) * 10)
20 DEF FNW(A,B) = A * B/2 + (A - B)
30 X = FNR2
40 PRINT X
50 Y = FNW(X,4)
60 PRINT Y
70 END
```

The two values that are DEFined as A, and B will be set equal to the corresponding values of X and 4. A will equal X and B will equal 4. The DEF statement will then use these numbers to calculate the (exp) part of the statement, and then Y will be made equal to this value. Finally, the value will be printed in line 60. You may wish to copy a program like this and experiment with it to see how it works.
press ENTER?

```
10 DEF FNR2 = INT(RND(0) * 10)
20 DEF FNW(A,B) = A * B/2 + (A - B)
30 X = FNR2
```

```
40 PRINT X
50 Y = FNW.X,4)
60 PRINT Y
70 END
```

What value would be passed to B in the second DEF statement?

A 4
B The value of X
C Cannot tell - not enough information

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

```
10 DEF FNX(A$) = LEN(A$)
20 A$ = "This is a function"
30 X = FNX(A$)
40 PRINT X
```

Here is an example of passing a string in a user function.
In all the examples that we have shown, we have always told
the computer when we wanted to invoke a user function by using
the prefix 'FN(var)'. That is the only way to get to your
user function.

press ENTER?

```
10 DEF FNX(A$) = LEN(A$)
20 A$ = "This is a function"
30 X = FNX(A$)
40 PRINT X
```

Note in the example above, that we passed a string within a
numeric variable, but that in the DEF statement, we set the
numeric variable equal to LEN(string). LEN(string) is an
example of yet another type of FUNCTION that we will study in
the next lesson. LEN(string) returns a numeric value equal to

```
the LENGTH of the string that is within the parenthesis.
Don't worry about it now, we will cover it in the next lesson.
However, because it is a NUMERIC value, its type matches the
DEF statement, and therefore it is valid.

press ENTER?



What kind of function do you create with the DEF statement

A  USER
B  LIBRARY
C  EITHER USER or LIBRARY
D  DEFINED FUNCTIONS

ENTER the letter opposite the correct answer? A

CORRECT ! - but that was an easy one, try the next question if
            you think you are up to it!
                 (we know you are - we're just trying
                  to lighten things up a bit!

press ENTER?



Is the following statement valid?

10 DEF FN88(A,B) = A * B

A  Yes, to call on it simply put FN88 somewhere in your program
B  No, the terms A and B are not defined and will cause errors
C  No, 88 is not a valid variable for this position
D  No, the syntax is good, but the beat is poor, you can't
       dance to it.

ENTER the letter opposite the correct answer?

WRONG - the correct answer is C

press ENTER?
```

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

You have completed this lesson and now you can go to the test
If you wish to review parts of this lesson, type in an 'R'.
If you wish to continue to the test, type in a 'C'.

Enter your choice (R or C)? C

Going to test number 5 - wait patiently please

FINAL TEST (lesson 5)

This test consists of 10 questions, you must get 70 percent
of them correct to pass. (that's 7 right out of the 10 ques-
tions). Use only capital letters in your answers, don't
include extra spaces or letters. If you successfully complete
the test, you can go on to the last lesson!
GOOD LUCK

press ENTER to continue?

Type in the order in which the following lines will be executed
Leave one space between each line number - ie, if the execution
sequence is ten, twenty and thirty, then type in 10 20 30

10 GOSUB 100

```
30 PRINT "DONE"
40 END
1000 RETURN

Enter the sequence now? 10 1000 20 30

CORRECT

press ENTER?
```

What will happen when you input then number 4 to the following program?

```
10 INPUT A
20 ON A GOSUB 1000,2000,3000
30 END
.... rest of program is not important
```

A Nothing
B Subroutine 3000 would be called
C Subroutine 1000 would be called
D The program would end

ENTER the letter opposite the correct answer? D

CORRECT

press ENTER?

How many RETURNS can you have in a subroutine?

A One
B As many as you want, but they should be kept to a minimum
C One for every GOSUB
D No more than the amount of memory available

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?

What is wrong with this program?

```
10 INPUT A
20 GOSUB 1000
30 END
1000 IF A <= 10 THEN RETURN
10005 IF A = 11 THEN RETURN
1010 GOTO 10
```

A Nothing
B One of the possible exits from the subroutine is incorrect
C There are too many RETURN statements
D The IF statements aren't allowed in a subroutine like this.

ENTER the letter opposite the correct answer? A

WRONG the correct answer is B
If A is greater than 10, then line 1010 causes
the subroutine to branch into the main program.
That will eventually cause the computer to get
mixed up and BOMB. See part 1, Subroutines.

press ENTER?

For the next few questions, you should be sure you have your
BASIC manual available so you can figure them out correctly.
All the questions are from part 2 of the lesson, and your
BASIC manual.

press ENTER?

What is the value of the following statement?

SQR(25

```
A  5
B  625
C  the natural logarithm of 25
D  100

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?



What type of functions would be written by you

Type in your answer using all capital letters. Do NOT append
the word FUNCTION on the end of your answer.

Type in your answer now? USER

CORRECT

press ENTER?



What is the term you would use to define a USER function.
(Hint: It's two words. DON'T include a variable, leave a space
between the two words)

ENTER your answer now? DEF FN

CORRECT

press ENTER?



10 DEF FNX = 10 * 2
20 Y = ?????

What is the term that would complete line 20 if we wished to
call on the user function in line 10?
```

ENTER your answer now? FNX

CORRECT

press ENTER?

Which of the following functions is valid?

A DEF FNX(Z,M) = Z * M/2 + A$
B DEF FN10(Z,M) = Z * 3 + Z + M
C DEF FNY(A$,X) = LEN(A$) + X
D DEF FNX = SQR(-4)
E DEF FNX = LOG(0)

ENTER the letter opposite the correct answer? C

CORRECT

press ENTER?

What is the proper statement to exit from a subroutine?

ENTER your answer now? RETURN

CORRECT

press ENTER?

You have finished the test. out of 10 possible correct answers
you scored 9 .

YOU HAVE PASSED
YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:

    part 1. SUBROUTINES

press ENTER?

Do you want your score recorded on a permanent file?

A YES
B NO

Which? B

You are now qualified to go to LESSON 6.
You may return to the MENU or receive your homework.

Do you want your homework assignment (Y/N)?

Break in 3830
Ready
SYSTEM"RESET +00"

```
Ready
RUN"LESSON6
```

LESSON: BASIC 6 VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON: Less than one hour

AUTHOR: Capt Danny J. Creagan
        Air Force Institute of Technology

OBJECTIVES: To teach the student about string functions
           and the Microsoft Editor

Press the ENTER key to continue?

LESSON 6

This is the first part of a two part lesson
It is divided into the following sections.

1) Introduction 4) Concatenation
2) String Assignment 5) String Functions
3) String ARRAY's

A I'm taking this part in its entirety.
B I wish to review selected areas.
C I want to go to the second part.
D I want to return to the Menu.

Press either capital A, B, C, or D and then press ENTER? A

INTRODUCTION

This is your last lesson! Congratulations! If you have taken the previous five lessons, you should be feeling a little more comfortable with Microsoft BASIC by now.

In this lesson we will cover STRING functions first, and then we will learn about the Microsoft Editor (in the second half).

As we learned before, a BASIC string is one or more alphanumeric characters that are treated as a single collection of data. Using the concepts in this chapter, you can perform many of the same types of operations on STRINGs that you can perform on numeric data.

press ENTER?

INTRODUCTION

As a small review, you should remember that string data can be designated in two ways. You can assign your data to a string variable, or you can enclose the data in quotes. Here are two examples.

```
10 PRINT "This is one way to designate a string "
20 V$ = "Another way is to put it in a variable and print it"
30 PRINT V$
```

Line 10 prints the string data immediately, and line 20 loads the variable V$ with the data. V$ can then be printed whenever we want it.

press ENTER?

INTRODUCTION

```
10 PRINT "This is one way to designate a string"
20 V$ = "And this is another"
30 PRINT
40 PRINT V$
```

RUN

This is one way to designate a string

And this is another

Here is another variation of our little program. Note that
'And this is another' is not printed until line 40 is executed.

press ENTER?

INTRODUCTION

Remember, in this lesson, as in all of our lessons, you should
have either a good BASIC manual handy, or you should have
an experienced programmer around to help you with difficult
problems.

In some of the answers you will need to be sure you use the
correct case (either uppercase or lowercase), so be sure to
read all the questions carefully.

Get out your manual, or programmer, now, and let's enjoy BASIC!

press ENTER?

STRING ASSIGNMENT

As we showed you in the introduction, you assign strings to
a variable and then you can print the variable anywhere in the
program. That makes it easier to write long program lines,
because you don't have to keep typing in the text every time
you want to use the string data.

You can assign data to strings using any of the statements we
used to assign numeric data to numeric variables. LET, READ,
and INPUT are all used with string assignment (LET is optional
just as it is with numeric data).

press ENTER?

Are these statements legal? (assume the program is just for demonstration, and that B$ is blank)

```
10 READ A$
20 INPUT A$
30 LET A$ = B$
40 A$ = "NOW IS THE TIME"
50 DATA "NOW IS THE TIME"
```

A No, the LET statement in line 30 is illegal
B No, the string assignment in line 40 is illegal
C No, you cannot read data into a string (line 10 is bad)
D Yes, all statements are legal

ENTER the letter opposite the correct answer?

WRONG - all these assignments are legal

press ENTER?

String Assignment

```
10  READ A$,B$
20  PRINT A$
30  A$ = B$
40  PRINT A$
50 DATA "THE GRINCH","IS COMING"
RUN

THE GRINCH
IS COMING
```

Notice that A$ was converted to B$, and all the data was printed out by using just A$ in print statements.

press ENTER?

String Assignment

```
10  READ A$,B$
20  PRINT A$
30  A$ = B$
40  PRINT A$
50 DATA "THE GRINCH","IS COMING"
```

Notice the dollar sign is always included with a string variable. The dollar sign tells the computer to treat the variable as a string instead of as a numeric. Also, whenever you assign data to a string, it must either be another string or it must be enclosed in quotes.

press ENTER

Assign NOW IS THE TIME to a string variable called NN$ and use line number 10 as your statement number.

Put one blank between terms.

ENTER your answer now ?

WRONG - the correct answer is 10 NN$ = "NOW IS THE TIME"
(you could have also answered 10 NN$ = " NOW IS THE TIME ")

press ENTER

You can also INPUT# string data from an external file. (the following program assumes that a file named TEST was previously created on disk)

```
10 OPEN"I",1,"TEST"
20 IF EOF(1) THEN END
30 INPUT#1,A$
40 PRINT A$
50 GOTO 20
```

A program such as this is used to read in your name when you

take your test at the end of each lesson. The original is enhanced a little, but the BASIC idea is the same.

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

STRING ASSIGNMENT

As we showed you in the introduction, you assign strings to a variable and then you can print the variable anywhere in the program. That makes it easier to write long program lines, because you don't have to keep typing in the text every time you want to use the string data.

You can assign data to strings using any of the statements we used to assign numeric data to numeric variables. LET, READ, and INPUT are all used with string assignment (LET is optional just as it is with numeric data).

press ENTER?

Are these statements legal? (assume the program is just for demonstration, and that B$ is blank)

```
10 READ A$
20 INPUT A$
30 LET A$ = B$
40 A$ = "NOW IS THE TIME"
50 DATA "NOW IS THE TIME"
```

A No, the LET statement in line 30 is illegal

```
B  No. the string assignment in line 40 is illegal
C  No. you cannot read data into a string (line 10 is bad)
D  Yes. all statements are legal

ENTER the letter opposite the correct answer?

WRONG - all these assignments are legal

press ENTER?
```

String Assignment

```
10  READ A$,B$
20  PRINT A$
30  A$ = B$
40  PRINT A$
50 DATA "THE GRINCH","IS COMING"
RUN

THE GRINCH
IS COMING
```

Notice that A$ was converted to B$, and all the data was printed out by using just A$ in print statements.

press ENTER?

String Assignment

```
10  READ A$,B$
20  PRINT A$
30  A$ = B$
40  PRINT A$
50 DATA "THE GRINCH","IS COMING"
```

Notice the dollar sign is always included with a string variable. The dollar sign tells the computer to treat the variable as a string instead of as a numeric. Also, whenever you assign data to a string, it must either be another string or it must be enclosed in quotes.

press ENTER^

Assign NOW IS THE TIME to a string variable called NN$
and use line number 10 as your statement number.

Put one blank between terms.

ENTER your answer now ?

WRONG - the correct answer is 10 NN$ = "NOW IS THE TIME"
     (you could have also answered 10 NN$ = " NOW IS THE TIME ")

press ENTER?

You can also INPUT# string data from an external file.
(the following program assumes that a file named TEST was
previously created on disk)

```
10 OPEN"I",1,"TEST"
20 IF EOF(1) THEN END
30 INPUT#1,A$
40 PRINT A$
50 GOTO 20
```

A program such as this is used to read in your name when you
take your test at the end of each lesson. The original is
enhanced a little, but the BASIC idea is the same.

press ENTER^

which do you wish to do^

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

String ARRAYs

You can assign string data to arrays in the same way as you assign numeric data to arrays. Nearly all the rules are the same. The following is an example.

```
10 CLEAR 2000
20 DIM A$(100)
30 X = 0
40 X = X+1
50 INPUT "ENTER up to 99 strings, ENTER 'END' to stop";A$(X)
60 IF A$(X) = "END" THEN END
70 GOTO 40
```

press ENTER?

```
10 CLEAR 2000
20 DIM A$(100)
30 X = 0
40 X = X+1
50 INPUT "ENTER up to 99 strings, ENTER 'END' to stop";A$(X)
60 IF A$(X) = "END" THEN END
70 GOTO 40
```

Notice the CLEAR statement. Remember that you normally have only 50 - 100 characters of string space available, and if you are going to need more, you need to tell the computer. Also, note the DIM statement - we need declare our array size if it is over 10

press ENTER?

```
10 CLEAR 2000
20 DIM A$(100)
30 X = 0
40 X = X+1
```

```
50 INPUT "ENTER up to 99 strings, ENTER 'END' to stop";A$(X)
60 IF A$(X) = "END" THEN END
70 GOTO 40
```

The variable X acts as a counter to reference the proper pocket of the string array. The string array is referenced exactly like the numeric array. Note that this program will only read in the data. If you want to print it out, you will have to add some more statements on the bottom of the program.

press ENTER?

What is the CLEAR statement for in BASIC?

A To clear extra number space for the computer
B To clear extra string space
C To zerize all number variables
D To clear the screen
E To help the programmer understand more clearly

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?

String ARRAYs

```
10 CLEAR 2000
20 DIM A$(100)
30 X = 0
40 X = X+1
50 INPUT "ENTER up to 99 strings, ENTER 'END' to stop";A$(X)
60 IF A$(X) = "END" THEN END
70 GOTO 40
```

The rule for the DIM statement is the same as for numeric arrays. What is the maximum size of one leg of a string array if you don't use the DIM statement ... type ? for answer now?

```
WRONG - the max size of an array without a DIM statement is 10
        this is an important concept. You may wish to review
        lesson 3 before you go to the next section.

press ENTER?



How would you find out what was in the fifth pocket of the
single dimension array A$(X)?

A  PRINT A$(X)
B  PRINT A$(5)
C  PRINT A$
D  READ A$(X)

ENTER the letter opposite the correct answer? B

CORRECT

press ENTER?



Which do you wish to do?

A  Continue on
B  Review this section again

press the letter opposite the correct answer and press ENTER? A



                    Concatenation

You may link two strings together by using the 'plus' symbol.
For example:

10 A$ = "where"
20 B$ = "Some"
30 C$ = B$ + A$
40 PRINT C$  :'You could have said 'PRINT B$ + A$' too.
RUN
```

somewhere

In this case, the '+' symbol served to 'add' the two strings
together and create another string.
press ENTER?

What is the output of the following program?

```
10  A$ = "FLASH"
20  B$ = "DANCE"
30  PRINT A$ + B$
```

A FLASH
B FLASH
   DANCE
C FLASHDANCE
D DANCEFLASH

ENTER the letter opposite the correct answer? C

CORRECT

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

String Functions

For this section you will definitely need your BASIC manual, so
get it out now.

As with arithmetic functions, there are STRING functions. STRING functions are used to manipulate or explore the contents of a string. On the next screen there are several examples of STRING functions. We will go over several of these, but you will not have to memorize them. Rather, you should understand that if you need to access or modify any kind of string, you can probably find a string function that will do the job for you. String functions can be used as part of USER functions as you saw in lesson 5.

press ENTER for some examples of string functions?

String Functions

1) ASC(string)
2) CHR$(exp)
3) FRE(string)
4) INKEY$
5) LEN(string)
6) MID$(string,position,length)
7) RIGHT$(string,length)
8) LEFT$(string,length)

At first glance, these functions look like a lot of GREEK. In fact, they look pretty bad at second glance! However, they really are pretty easy to use, once you understand them. The best way to learn how to use them is to make a short program and use them one at a time until you see what they do. We will do over examples of a couple to help you catch on.

press ENTER?

Is the following statement TRUE or FALSE?

String functions are used to manipulate data within string variables.

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

String Functions

10 A$ = "My aching fingers"
20 PRINT ASC(A$)

ASC(string) is a function that returns the ASCII code of the first character of the string. ASCII stands for "AMERICAN STANDARD CODE for INFORMATION INTERCHANGE. Look up the ASCII code for the first letter of A$ in your BASIC manual. What is it? You should have found it to be 77 decimal.
When your computer writes data files to disk, it usually writes them in ASCII code, one letter at a time. This function has use when you are trying to convert characters to their number equivalent.

press ENTER?

String Functions

PRINT CHR$ 77

CHR$ (x) returns the opposite of the ASC(string) function. It returns a character equivalent of decimal 77. Which is "M".

press ENTER?

What is the output of the following program?

10 PRINT ASC("A")
20 PRINT CHR$(66)

A A
   66
B 66
   B

```
C  A x
D  B
   C

ENTER the letter opposite the correct answer? B

CORRECT - GREAT!

press ENTER?
```

String Functions

```
10 A$ = "My aching fingers"
20 PRINT LEN(A$)
```

LEN(string) is a function that returns the length of the string that is in parenthesis. In this case it should return 17.

What is the value of LEN("TOM SWIFT")?

```
A  11
B  8
C  1
D  9

ENTER the letter opposite the correct answer? A

WRONG - the correct answer is D

press ENTER?
```

Now you have to do some work for yourself. What will be the output of the following program?

```
10 A$ = "My aching fingers"
20 PRINT LEFT$(A$,2)

A  M.
B  M
C  My aching
```

0 Nothing will be output

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

```
10 A$ = "My aching fingers"
20 PRINT MID$(A$,4,6)
```

ENTER the output of this program? aching

CORRECT - I'm glad to see you use the book!

press ENTER?

```
10 A$ = "My aching fingers"
20 PRINT RIGHT$(A$,4)
```

ENTER the output of this program? @ERs

CORRECT - good job

press ENTER?

String Functions

Here's an interesting function. It's called INKEY$ and it strobes your keyboard ONCE and if a key is depressed, it returns the character that was pressed. Here is an example of how to use it.

```
10 IF INKEY$ = "S" THEN END
20 GOTO 10
```

If you type this program in exactly as shown, and RUN it, it

will keep running until you press the 'S' key. Try it when you
are done here.

press ENTER"

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? a

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

You have finished the first part of lesson 6. If you wish to
review this part, type in 'R', if you want to continue to the
next half, type in 'C'

ENTER an R or a C? C

Going to the next part, please standby

LESSON 6

This is the second part of a two part lesson
It is divided into the following sections.

1) Introduction   4) DO(elete)

2) Starting (EDIT/exit) 5) X(tend line)
3) nSPACEBAR, nS(earch) 6) nC(hange), I(nsert)
7) TEST

A I'm taking this part in its entirety.
B I wish to review selected areas (or take the test).
C I want to go to the first part.
D I want to return to the Menu.

Press either capital A B, C, or D and then press ENTER? A

Introduction

This part of the computer assisted instruction program has nothing to do with BASIC. Instead, it is about the Microsoft Editor which allows you to edit BASIC program statements so you don't have to retype a whole BASIC line just because of one typo. Using the editor makes it very easy to alter the line.

Until you get familiar with the editor, you may wish to make a little 'cheat sheet' so you can have the commands available for quick reference. Throughout this part you should have your manual open to the editor portion so you can follow along.

press ENTER?

INTRODUCTION

The object of this half will be to get you familiar with the editor's key commands. The lesson will not teach you all the commands available. However, the core of knowledge it gives you will let you start editing BASIC programs. For some of the more sophisticated commands, you should refer to your Microsoft Manual.

press ENTER? a

Starting

The editor is line oriented , meaning that you operate on one line at a time (and not on a screen of data like you do with a word processor).

Throughout this lesson, we will be using one example line to illustrate all the commands. That line is listed below.

10 FOR X = 5*to77 :PRINT X: NEXT I

The colons form what is called a MULTI STATEMENT line. Each time a colon is entered, the computer treats the data following it as a new line. Therefore, on the above line we have three statements. Obviously, there are several errors in the line.
press ENTER"

Starting

10 FOR X = 5*to77 :PRINT X: NEXT I

If we wanted to EDIT this line, we would type in the word EDIT, followed by the line number. In this case, we would type in EDIT 10. You may enter the EDITor in other ways, but for this lesson we will always use the EDIT (line number) syntax. Always access the editor from the IMMEDIATE mode.

If you want to EXIT the EDITor after you are done editing, then you just press ENTER. Pressing ENTER from the EDITOR mode updates the line, and puts you back in IMMEDIATE mode again.

press ENTER"

Starting

10 FOR X = 5*to77 :PRINT X: NEXT I

EDIT 10
10

In the above example, we typed in the word EDIT 10, and the computer put us in the EDIT mode. Note that the line number we are EDITing appeared. The cursor would normally be flashing just to the right of the line number.

press ENTER?

Type in the command to edit line 10

ENTER the command now? EDIT 10

GOOD .... now
you are in the EDIT mode. what key do you press to exit the EDIT mode? ... Press the key now ... DON'T PRESS 'BREAK or 'CONTROL' C'

EDIT 10

10 FOR X = 65TO77 :PRINT X: NEXT I

Note how the rest of the line appeared after you pressed ENTER. That's the way it would happen if you were actually doing it.

press ENTER?

SPACEBAR, S(earch)

To move the cursor over the line you are editing, you simply press the spacebar, and it will move over one character.

For example, let's say you entered the edit mode and the cursor is just on the right of the line number. Go ahead and press the spacebar until you get to the end of the line.
After the line is printed, press spacebar once more to go on:

10 FOR X = 65TO77 :PRINT X: NEXT I
EDIT 10
10 FOR X = 65TO77 :PRINT X: NEXT I

GOOD .. Do you see how that worked?
Normally, your cursor would also be flashing, and you would still be in EDIT mode, with your cursor on the last character.

press ENTER?

nSPACEBAR, nS(earch)

Notice the lowercase 'n' in the title to this section? That refers to a feature of the editor that allows you to position the cursor 'n' spaces to the right. That way you won't have to pound away at your spacebar to get to the 200th character of a long line. All you do is press the number of characters you want to 'spacebar' over and then press the spacebar.
In this example, you are in the edit mode and you want to go over a few spaces.

Press a number (up to 255) , and then press the spacebar!

```
10 FOR X = 1 to 77 :PRINT X: NEXT X
EDIT 10
10 FOR X
```

WHEW That was quick!.. If you want to do it again, ENTER a 'Y'
else ENTER an 'N'? N

nSPACEBAR, nS(earch)

If you entered a bigger number that your line length, then the computer just defaulted to the maximum line length.

Also, if you were actually using the editor, you would stay in the EDIT mode, with the cursor over the nth character, after you pressed the spacebar.

OK, we managed to move the spacebar around a little, what if we wanted to find a specific character in the line, and we weren't quite sure exactly how far down the line it was?
In that case we would use the nS(earch) feature.

press ENTER?

nSPACEBAR, nS(earch)

With the Search feature, right after you type in EDIT (line number), you can drive the cursor to any letter in the line and if the letter is not in the line, then the editor de-faults to the end of the line. The editor only searches to the right of the cursor.

For example, you just typed in EDIT 10 and you wish to find the letter X. All you do is press 'S' and then press X. (The editor recognizes upper and lower case, so be sure the case is right). Go ahead and do it now. (You may search for any letter, just type S first, then type the letter)

```
10 FOR X = 5*to77 :PRINT X: NEXT X
EDIT 10
10 FOR X = 5*to77 :PRINT X: NEXT
```

WOW ! Another quick one... ENTER 'Y' to do again, else ENTER an 'N'? N

nSPACEBAR, nS(earch)

You will have noticed that the cursor stopped BEFORE the character that you were searching for. That is what is supposed to happen. In addition, if you asked to search for a non-existent character, the computer printed the whole line.

No doubt, you will also have noticed that there is a lower case 'n' in the title nS(earch). It simply means that you can search for the nth occurance of the specific character. For example, if we were in the EDIT mode for line 10, and we wanted to find the second occurance of the letter 'R' then we would type '2SR' and the cursor would skip over to the R in PRINT X.

press ENTER

^SPACEBAR^ ^Search^

Here is an example of the ^Search^ feature.

10 FOR X = 65TO77 :PRINT X: NEXT I
EDIT 10
10 I'Now type ISP to find 2nd occurance of R in 10

1. FOR X = 65TO77 IF

The cursor would stop just before the nth occurance of R.

press ENTER

What would you type to find the 2nd occurance of the character
"I" in a line you were editing (because you are already in the
EDIT mode.)

ENTER your answer now" ISP

CORRECT - GREAT

press ENTER

Which do you wish to do"

C Continue or
R Review this section again

press the letter corresponding the correct answer and press ENTER" R

^Delete^

Hopefully, you are catching on to the way the boys and girls at Microsoft are doing things, and won't have too much trouble with this command. It does what it looks like it does. It deletes characters, one at a time, or 'n' at a time.

For example, if you are in the EDIT mode for line 10 and you want to delete the NEXT character, then just press D and the character will be enclosed in exclamation marks. The exclamation marks indicate that if you don't change things, then the new line will not have the character in it.

press ENTER

An example of the D(elete) command would be:

```
10 FOR X = 6*1TO77 :PRINT X :NEXT X
EDIT 10
I                      :'now say we want to delete the '6*' chars
                       :'all we do is put the cursor to the left
                       :'of the two characters and press D twice
10 FOR X = !6*!        :'and it would look like this
                       :'then we would press ENTER and we would

10 FOR X = 1TO77 :PRINT X :NEXT X       : have this
```

Study this example and read the appropriate paragraph in your manual.

press ENTER

nD(elete)

```
10 FOR X = 6*1TO77 :PRINT X :NEXT X
```

Now it is your turn. You have to EDIT the above line so that the characters '6*' are deleted. You must delete them one at a time (as we showed you in the previous example). First type the appropriate command for editing line 10, then move the spacebar over to the appropriate place, then delete the two offending characters, then press ENTER (use upper case

ENTER the first command now?

WRONG - you should type in EDIT 10 first!

ENTER the first command now? EDIT 10

10 FOR X = !a!!*!to77 :PRINT Y :NEXT I

The line in the computer's memory would now look like this:
10 FOR X = to77 :PRINT Y :NEXT I

GREAT ... If you want to do it again, press 'Y' else press 'N'
ENTER your choice now (Y or N)? N

nD(elete)

For the example, we protected you from mistakes by ignoring some commands, and telling you what you did wrong for others. If you really are editing a line, be sure you press the right buttons, because you won't get warning messages! However, you seldom ruin what you have done, the editor is very forgiving, it usually leaves you something, even when you make a boo-boo!

As you may have guessed, the 'n' symbol in the title indicates that you may ... e 'n' characters at a time. If you press a number and press D, then that is how many characters will be deleted.

press ENTER

10 FOR X = a*to77 :PRINT Y : NEXT I

If we wanted to delete 'a*' all a once, we could position the cursor to just before the 'a' and type in 2D and both characters would appear like this: !a*

Note the exclamation marks, in this case, are around both characters. When you get through here, practice with some

lines you have arbitrarily made up. You will see that this
command can be very handy.

press ENTER

which do you wish to do?

1 Continue on
2 Review this section again

press the letter opposite the correct answer and press ENTER 1

Extend line

This is one of the easiest, and most useful of the commands
It allows you to start up at the end of a line, just as if you
never pressed ENTER.

First, you go to EDIT mode, then you press X, when you do, you
will see the whole line displayed, and you can add anything
on to the END of it. Try it now. First, type the command to
get into EDIT mode for line 1, then press X, then type in
anything you want Most micros allow a maximum of 249
characters . Then press ENTER. Do it now.

ENTER the exact command? EDIT 1

1 FOR X = 0 TO 255 :PRINT X :NEXT X

Our new line, which was the sum of the old line 1 plus the
data you typed in, is now this:

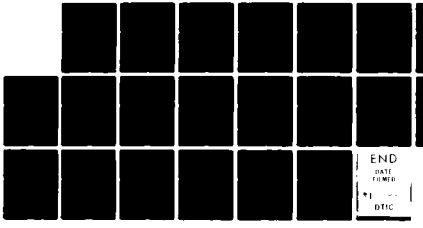
1FOR X = 0 TO 255 :PRINT X :NEXT X

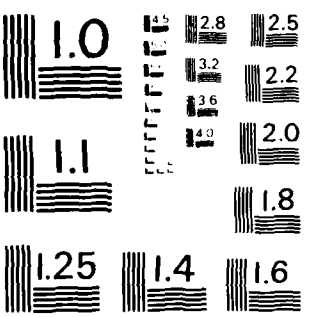
Neat huh? You will find yourself using this command the most,

As usual, we protected you from making mistakes. In the

AD-A134 386 COMPUTER ASSISTED INSTRUCTION IN BASIC(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS D J CREAGAN 28 SEP 83 AFIT-LSSR-29-83 6/6

UNCLASSIFIED F/G 9/2 NL





allowing you to execute the X(tend) command. Remember, you will have much more freedom if you really are in the editor mode. In fact, in the real editor, you can use the back arrow to wipe out the end of the line you are editing, and replace the old data with new stuff. Be sure to practice this and you will grow to love it -- ...well, maybe just like it a lot.

press ENTER

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER A

(C)hange) & (I)nsert)

Now for the meat of this half: we are going to change the nasty errors in our trial statement, and then Insert some correct figures.

FOR X = to7 :PRINT Y :NEXT I

Above is our line (with the 'S*' characters missing - we deleted them in the (D)elete) section). Let's say we want to change 'NEXT I' to 'NEXT X' and 'PRINT Y' to 'PRINT X'. In addition, we want the value of X to start at 1 in the FOR NEXT loop, i.e. we want to insert a 1 just before the to7

press ENTER to start our EDITing?

(C)hange) & (I)nsert

FOR X = to7 :PRINT Y :NEXT I

First we'll change 'PRINT Y' to 'PRINT X'

To use the nC(hange) command, get into the EDIT mode and position the cursor to JUST BEFORE the character to be changed then press 'C' and press the new character, then press ENTER. Try it now. Get to EDITor, SPACE over to just before the ' and type in 'C', then type in an X (we want to swap Y with X) and finally, press ENTER (use capitals)

ENTER the first command at the bottom of the next line
10 FOR X = to77 :PRINT Y :NEXT I
EDIT 10

10 FOR X = to77 :PRINT X :NEXT I

How's that for class? Remember, we protected you from mistakes. The reg. editor will do whatever you tell it, even if it is wrong . But you knew that, didn't you?

ENTER a 'Y' if you want to do this again, else ENTER an 'N' N

nC.hange) & I.nsert)

As with the other commands, the 'n' in nC.hange) designates how many characters are affected by the command. If you want to change 10 characters, then you would type '10C' in the EDIT mode, and you would then HAVE to change the next 10 characters.

What would you type if you were in the EDIT mode and your cursor was just before a block of 4 characters that you wanted to change to 'XXXX'?

ENTER your answer now' 4CXXXX

GREAT Now you have the idea

press ENTER

nC.hange & I.nsert)

10 FOR X = to77 :PRINT X :NEXT I

we would change the 'I' to an 'X' in the same way, but, to save time, we'll envoke some magic, and change it now so we can get to the I(nsert) command! READY? zzzzzZZZZHAAPPP! . . POOF . . There, it's changed now. Look below.

```
10 FOR X = to77 :PRINT X :NEXT X
```

How would you like to have THAT editor at your command? we'll now get to the I(nsert) command. Remember, we want to Insert a 1 just before the 'to77'.

press ENTER for the Insert example?

nC(hange) and I(nsert)

```
10 FOR X = to77 :PRINT X :NEXT X
```

To use the I(nsert) command, you first get into the EDIT mode and then place the cursor to just before the character you want to insert the data in front of.

In this case, we get into the EDIT mode, then we SPACE over to just before the 'to77' and then we type an 'I' for I(nsert). After the 'I' command we want to put in a 1, but we COULD type in as many characters we want .... until we press ENTER. At that time, all our changes are made and we are returned to the IMMEDIATE mode.

press ENTER to start the example?

Remember, first type EDIT 10, then space over to just before the 'to77', then type 'I', then type a 1, then type ENTER.

```
10 FOR X = to77 :PRINT X :NEXT X
```

ENTER the first command? EDIT 10

```
10 FOR X = 1to77 :PRINT X :NEXT X
```

ENTER a 'Y' to do this again, else ENTER an 'N'? N

nC(hange) & I(nsert)

With some of the same magic we envoked before, we will also put spaces in the proper places of the test line. (again, you would normally use your nC or I command to fix up the line but I feel pretty magical today, so I want to do it.

zzzZZZAAAAAAPPPP <<< POOF >>>......CRAASSSHHHH .. tinkle !(FORGROUND)(TWIXAGLFUG)(()***sluurrrrrpp...........oooops!

Oh well, we'll leave it up to you to do in your practice sessions.

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

press the letter opposite the correct answer and press ENTER? A

You have finished the lesson and you can now take the test.
If you wish to review parts of the lesson, ENTER an 'R'
else, if you want to continue to the test ENTER a 'C'

ENTER your choice now (R or C)? C

Going to test. Please wait one moment.

FINAL TEST (lesson 6

This test consists of 10 questions. You must get 70 percent of them correct to pass. (That's 7 right out of the 10 questions). Use only capital letters in your answers, don't include extra spaces or letters.

GOOD LUCK!

press ENTER to continue?

What is wrong with the following statement?

```
10 A$ = "Hi!":B$ = A$ :B$ = LEFT$(B$,1) :PRINT B$
```

A Multi Statement lines are not allowed
B The assignment of values between A$ and B$ are not valid.
C The strings were not initialized
D Nothing

ENTER the letter opposite the correct answer? D

CORRECT

press ENTER?

```
10 A$ = "Hi!":B$ = A$ :B$ = LEFT$(B$,1) :PRINT B$
```

What is the output of the above program? (You may use your BASIC manual to look up terms)

ENTER the output now EXACTLY as it would appear? Hi!

WRONG - the correct answer is H
        If A$ = Hi and B$ = A$, then the left character of B$ is an H. See part 1.

press ENTER?

What is wrong with the following program.

```
10 FOR X = 1 to 20
20   A$(X) = "0"
30 NEXT X
```

A The array is not dimensioned properly
B You cannot address a single dimensioned array with a loop
C The `0` should not be enclosed in quotes
D Nothing

ENTER the letter opposite the correct answer? a

WRONG - the answer is A (it should be DIMensioned to 20)
        See part 1. String Arrays.

press ENTER?

What is the output of the following program?

```
10 A$ = "Hi'";B$ = A$ :B$ = LEFT$(B$,1) :PRINT B$
20 B$ = A$ + B$
```

ENTER your answer EXACTLY as it would appear? Hi'H

CORRECT

press ENTER?

What is the output of the following program?

```
10 A$ = "SOMEWHERE":A$ = MID$(A$,1,4):PRINT A$
```

ENTER your answer EXACTLY as it would appear? SOME

CORRECT

press ENTER?

What is the command you would enter to edit line number 50 of a program?

ENTER your answer EXACTLY as it would appear? EDIT 50

CORRECT

press ENTER?

Assume you are in the EDIT mode. You wish to place the cursor over the second occurance of the letter R in your line.

What is the command you would use?

A 2SR
B 2DR
C 2FR
D 2CR
E 2 spacebar R

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

Assume your are in the EDIT mode

what is the command you would use to insert text starting where your cursor is now.

ENTER the command now? I

CORRECT

press ENTER?

Assume you are in the EDIT mode

What is the command you would use to drive the cursor to the
end of the line you are currently editing. (The command is
one letter long.)

ENTER the command now? X

CORRECT

Press ENTER?

Assume you have just finished a course in computer assisted
instruction in BASIC. What should you do?

A Quit trying, now that you know how
B Practice, practice, practice . . . . and enjoy, enjoy, enjoy
C Sell yourself as a national treasure
D Write a nasty letter to the author of the program

ENTER the letter opposite the correct answer? A

Obviously, we have failed to communicate. I'm reporting you to
the FBI for tax evasion and mail fraud.
Try AGAIN

Press ENTER?

Assume you have just finished a course in computer assisted
instruction in BASIC. What should you do?

A Quit trying, now that you know how
B Practice, practice, practice . . . . and enjoy, enjoy, enjoy
C Sell yourself as a national treasure
D Write a nasty letter to the author of the program

ENTER the letter opposite the correct answer? B

Don't try to butter me up, I know you're into masochism!

press ENTER?

Obviously, you get automatic credit for the last question.

It was nice doing business with you. So long!

press ENTER?

You have finished the test. out of 10 possible correct answers
you scored 9 .

YOU HAVE PASSED
YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:

  part 1. STRING ASSIGNMENT, STRING FUNCTIONS
  part 1. STRING ARRAYS

press ENTER?

Do you want your score recorded on a permanent file?

A YES
B NO

Which? B

You are FINISHED WITH THIS COURSE !!!

It was GREAT having you as a student - THANKS!
When you press ENTER you will be sent to the Menu from

```
where you can review other lessons or quit

For homework, you may wish to change your inventory program
so that it will handle string data.  That way you can include
the names of your furniture in your file.  However, we leave
that up to you. Hasta Luego!


press ENTER to return to MENU?

Break in 4250
Ready
system"reset #00"
```

APPENDIX D

HOMEWORK MODULES

TABLE OF CONTENTS

```
1000 GOSUB 1520
1010 PRINT"Your lesson 2 assignment is to write a program that requests the"
1020 PRINT"user to INPUT a number.  The program should check the number"
1030 PRINT"against numbers in a data statement to see if there is a"
1040 PRINT"match. If there is a match, then the program should tell"
1050 PRINT"the user.  Use the RESTORE and GOTO statements to keep the"
1060 PRINT"program running.  Save the program to disk."
1070 PRINT
1080 PRINT"You will use much of what we have learned when you make this"
1090 PRINT"program.  If you feel a little confused by the directions,"
1100 PRINT"you can get an example of a program that satisfies the"
1110 PRINT"requirements by turning on your printer and selecting the"
1120 PRINT"LIST PROGRAM TO PRINTER option below."
1130 PRINT
1140 PRINT"Which do you want to do?"
1150 PRINT
1160 PRINT"A   RETURN TO MENU"
1170 PRINT"B   LIST PROGRAM TO PRINTER"
1180 PRINT"C   RUN THE PROGRAM"
1190 PRINT
1200 INPUT"Enter your choice";T$
1210 IF T$ = "A" THEN RUN "MENU"
1220 IF T$ = "B" THEN PRINT:PRINT"Be sure to type in RUN when the printer is done.":LLIST 1250-1500
1230 IF T$ = "C" THEN GOSUB 1520:PRINT"Be sure to type RUN when program ends.":PRINT:INPUT"press
     ENTER to start";T$:RUN 1250
1240 GOTO 1140
1250 REM This program asks for a number, compares the number to a
1260 REM data list, and then tells the user if the correct number
1270 REM was selected.  It is not the only way to do the problem.
1280 REM
1290 REM Next line resets the data statement (see lesson 2, part 2)
1300     RESTORE
1310 REM Now we ask for the number (I use 1 to 20 to keep it simple)
1320     INPUT"Enter a number between 1 and 20 (enter -1 to quit)";N
1330     IF N = -1 THEN GOTO 1500
1340 REM Now we read a data point to see if we match (See lesson 2, part 2)
1350     READ D
1360     DATA 1,3,7,10,15,18,0
1370 REM Above DATA is arbitrary, I just made a few points up
1380 REM Now we compare the data point with the inputted number (see part 2)
1390     IF N = D THEN PRINT "MATCH - There is a data point that equals";N
1400 REM If we have a match, we should RESTORE our data and start over
1410     IF N = D THEN GOTO 1300
1420 REM If our data point = 0, then we are out of points to check, so
1430 REM we should RESTORE the data, tell the user there was no match
1440     IF D = 0 THEN PRINT "Your number does not match any of the data."
```

```
1450     IF D = 0 THEN GOTO 1500
1460 REM IF D doesn't equal 0 and it doesn't equal the present data
1470 REM point, then we read ANOTHER data point and check it all again.
1480     GOTO 1350
1490 REM
1500     END
1510 REM the following subroutine clears the screen
1520 FOR X = 1 TO 24
1530     PRINT
1540 NEXT X
1550 RETURN
```

```
 10 GOSUB 720
 20 PRINT"For your lesson 3 homework assignment, write a program"
 30 PRINT"that requests three numbers from the user, and puts the"
 40 PRINT"numbers in an array that has been DIMensioned to 3."
 50 PRINT"Finally, print the array to the screen so that the numbers"
 60 PRINT"appear in ASCENDING order. Use a FOR NEXT loop in your program."
 70 PRINT
 80 PRINT"I have made a sample program that satisfies the requirements. You"
 90 PRINT"may RUN it or list it to the printer to see how it works."
100 PRINT"If you choose choice B, be sure your printer is ready."
110 PRINT
120 PRINT"Which do you want to do?"
130 PRINT
140 PRINT"A   RETURN TO MENU"
150 PRINT"B   LIST PROGRAM TO PRINTER"
160 PRINT"C   RUN THE PROGRAM"
170 PRINT
180 INPUT"Enter your choice";T$
190 IF T$ = "A" THEN RUN "MENU"
200 IF T$ = "B" THEN PRINT:PRINT"Be sure to type in RUN when the printer is done.":LLIST 220-700
210 IF T$ = "C" THEN GOSUB 720:PRINT"You must type in RUN when the program
    ends":PRINT:INPUT"press ENTER to start";T$:GOSUB 720:RUN 220
220 GOTO 120
230 REM This program requests three numbers from the user and puts the
240 REM numbers in a three pocket array.  It then prints the array to screen
250 REM so that the numbers appear in ASCENDING order.
260 REM
270 REM First, we DIMension the array (see lesson 3 part 2)
280 REM
290     DIM A(3)
300 REM
310 REM Then we use a FOR NEXT loop to ask for three numbers (this is only
320 REM a way of doing this problem, you may want to do it differently).
325 REM  see lesson 3 part 1 for info on FOR NEXT)
330 REM
340     FOR X = 1 TO 3
350         INPUT"Enter a number";A(X)
360     NEXT X
370 REM
380 REM Now we print a general heading so that our output looks a little
390 REM better.
400 REM
410     PRINT
420     PRINT"These are your numbers in ASCENDING order":
430     PRINT
440 REM
```

```
450 REM We can sort the numbers in many ways, but the most straight
460 REM forward is to use IF statements
470 REM
480 REM  For this problem, it helps to figure out how many ways three
490 REM  numbers can occur. (the answer is 6 - 123.132.213.231.312.321)
500 REM  That means you can do it with six IF statements.(see lesson 3 part 2)
510 REM
520     IF A(1)<A(2) AND A(2)<A(3) THEN PRINT A(1);A(2);A(3)
530     IF A(2)<A(1) AND A(1)<A(3) THEN PRINT A(2);A(1);A(3)
540     IF A(3)<A(1) AND A(1)<A(2) THEN PRINT A(3);A(1),A(2)
550     IF A(1)<A(3) AND A(3)<A(2) THEN PRINT A(1);A(3);A(2)
560     IF A(2)<A(3) AND A(3)<A(1) THEN PRINT A(2);A(3);A(1)
570     IF A(3)<A(2) AND A(2)<A(1) THEN PRINT A(3);A(2);A(1)
580     END
590 REM
600 REM  Whew! that is quite a bunch of statements for just three data
610 REM  points!  For a more sophisticated sort, load the training report
620 REM  module and look at the bubble sort routine in it (use LOAD "REPORT").
630 REM  That routine will sort an array of ANY length, yet it is just
640 REM  10 statements long. The secret is in a clever use of FOR NEXT.
650 REM
660 REM  For more information on programs like that, check out one of the
670 REM  many books on BASIC in your library. You can do just about
680 REM  anything in BASIC.
690 REM
700 REM
710 REM the following subroutine clears the screen
720 FOR X = 1 TO 24
730     PRINT
740 NEXT X
750 RETURN
```

```
 10 GOSUB 310
 20 PRINT"For your lesson 4 homework, improve the program you made in"
 30 PRINT"lesson 3.  Write the array to disk.  Write another small program"
 40 PRINT"to read the data back from the disk.  THEN print the data in"
 50 PRINT"ASCENDING order."
 60 PRINT
 70 PRINT"I have made a sample program that satisfies the requirements.  You"
 80 PRINT"may RUN it or list it to the printer to see how it works."
 90 PRINT"If you choose choice B, be sure your printer is ready."
100 PRINT
110 PRINT"Which do you want to do?"
120 PRINT
130 PRINT"A   RETURN TO MENU"
140 PRINT"B   LIST PROGRAM TO PRINTER"
150 PRINT"C   RUN THE PROGRAM"
160 PRINT
170 INPUT"Enter your choice";T$
180 IF T$ = "A" THEN RUN "MENU"
190 IF T$ = "B" THEN PRINT:PRINT"Be sure to type in RUN when the printer is done.":LLIST 220-790
200 IF T$ = "C" THEN GOSUB 310:PRINT"You must type in RUN when the program
    ends":PRINT:INPUT"press ENTER to start";T$:GOSUB 310:RUN 220
210 GOTO 110
220 REM This program requests three numbers from the user and puts the
230 REM numbers in a three pocket array.  It then prints the array to disk.
240 REM The second part of the program reads the data back from disk and
250 REM prints it in ASCENDING order.
260 REM
270 REM First, we DIMension the array (see lesson 3 part 2)
280 REM
290     DIM A(3)
300 REM
310 REM  Then we open a file to print the data to disk (see lesson 4 part 1)
320 REM
330      OPEN"O",1,"TEST"
340 REM
350 REM
360 REM Then we use a FOR NEXT loop to ask for three numbers (this is only
370 REM my way of doing this problem.  You may want to do it differently).
375 REM (see lesson 3 part 1 for info on FOR NEXT)
380 REM
390 REM  I chose to write the array to disk as it was entered by the user
400 REM
410     FOR X = 1 TO 3
420         INPUT"Enter a number";A(X)
430         PRINT#1,A(X)
440     NEXT X
```

```
450 REM
460 REM  Now we close the file (see lesson 4 part 2)
470 REM
480     CLOSE
490 REM
491 REM  We'll stop the program here and let the user know what happened
492 REM
493      PRINT
494      PRINT"The data is on disk.  When you press ENTER the data will be"
495      INPUT"read back from disk and printed in order.  Press ENTER";T$
496 REM
500 REM Now we print a general heading so that our output looks a little
510 REM better.
520 REM
530     PRINT
540     PRINT"These are your numbers in ASCENDING order":
550     PRINT
560 REM
570 REM Open the file for input and read the three data points into arra. A
580 REM
590     OPEN"I",1,"TEST"
600     FOR I = 1 TO 3
610         INPUT#1,A(I)
620     NEXT I
630     CLOSE
640 REM We can sort the numbers in many ways, but the most straight
650 REM forward is to use IF statements (see lesson 2 part 2)
660 REM
670 REM  For this problem, it helps to figure out how many ways three
680 REM  numbers can occur. (the answer is 6 - 123,132,213,231,312,321)
690 REM  That means you can do it with six IF statements.
700 REM
710     IF A(1)<A(2) AND A(2)<A(3) THEN PRINT A(1);A(2);A(3)
720     IF A(2)<A(1) AND A(1)<A(3) THEN PRINT A(2);A(1);A(3)
730     IF A(3)<A(1) AND A(1)<A(2) THEN PRINT A(3);A(1),A 2)
740     IF A(1)<A(3) AND A(3)<A(2) THEN PRINT A(1);A(3);A(2)
750     IF A(2)<A(3) AND A(3)<A(1) THEN PRINT A(2);A(3);A(1)
760     IF A(3)<A(2) AND A 2)<A(1) THEN PRINT A(3);A(2);A(1)
770     END
780 REM
790 REM
800 REM the following subroutine clears the screen
810 FOR X = 1 TO 24
820     PRINT
830 NEXT X
840 RETURN
```

```
10 GOSUB 2270
20 PRINT"Your homework assignment for lesson 5 is GREAT!  You know enough"
30 PRINT"about computers to make a useful program that could ease your"
40 PRINT"workload.  Your homework assignment is:"
50 PRINT
60 PRINT"    Write a program that will write a record to disk of all the"
70 PRINT"    stock numbers of furniture that are in your office. (only use"
80 PRINT"    a few numbers to start, until you get it debugged!)"
90 PRINT
100 PRINT"    Next, write a program that will let you add a record onto"
110 PRINT"    the end of the sequential file of stock numbers."
120 PRINT
130 PRINT"    Finally, write a program that will search through the"
140 PRINT"    disk file for a specific STOCK CLASS of numbers and"
150 PRINT"    print them out to screen. (a stock class is the first 4 digits"
160 PRINT"    of the stock number)"
170 PRINT
180 INPUT"press ENTER";T$
190 GOSUB 2270
200 PRINT"HINT #1:"
210 PRINT
220 PRINT"This assignment requires you to use a number that exceeds 7 digits"
230 PRINT"and whenever you use a number that long in BASIC, the computer"
240 PRINT"always changes it to scientific notation (see your manual).  To"
250 PRINT"prevent that, you must declare the number that you write and read"
260 PRINT"from disk as double precision.  The command for that is: DEFDBL (var)"
270 PRINT
280 PRINT"10 DEFDBL X"
290 PRINT"20 X = 123456789                    10 X = 123456789"
300 PRINT"30 PRINT X                          20 PRINT X"
310 PRINT"RUN                                 RUN"
320 PRINT
330 PRINT"123456789                           1.234567E+08"
340 PRINT
350 PRINT"As you see in the example, X was not changed to scientific notation"
360 PRINT"when it was declared double precision."
370 PRINT
380 INPUT"press ENTER";T$
390 GOSUB 2270
400 PRINT"HINT #2"
410 PRINT
420 PRINT"In lesson 4, part 2, there was an example of how to update a"
430 PRINT"sequential file.  Here is a summary of that example."
440 PRINT
450 PRINT" 1. OPEN the file for sequential input"
460 PRINT" 2. OPEN another new file for sequential output"
```

```
470 PRINT" 3. INPUT a data point from the old file."
480 PRINT" 4. OUTPUT the data point to the new file."
490 PRINT" 5. When the old file is empty, add your new data to the end"
500 PRINT"    of the new file."
510 PRINT
520 PRINT"THIS IS NOT THE MOST EFFICIENT METHOD, BUT MOST STUDENTS FIND IT"
530 PRINT"THE EASIEST TO START WITH.  IN THE EXAMPLE PROBLEM, I WILL SHOW "
540 PRINT"YOU HOW TO DO IT MORE EFFICIENTLY!"
550 PRINT
560 INPUT"press ENTER";T$
570 GOSUB 2270
580 PRINT"HINT #3:"
590 PRINT
600 PRINT"To find the STOCK CLASS of an 11 digit  number, you would multiply"
610 PRINT"the stock number by .0000001 and take the integer of it.  The result"
620 PRINT"would be the stock class.  For example:"
630 PRINT
640 PRINT"stock number = 58955746431"
650 PRINT"58955746431 * .0000001 = 5895.5746431"
660 PRINT"INT(5895.5746431) = 5895 = stock class."
670 PRINT
680 INPUT"press ENTER";T$
690 GOSUB 2270
700 PRINT"All this seems like a tall order, but it really isn't too bad."
710 PRINT"Just take each program a step at a time.  Here is a brief summary:"
720 PRINT
730 PRINT" 1. Write a program that puts stock numbers into a disk file."
740 PRINT" 2. Write a program that puts stock #s on the end of the file."
750 PRINT" 3. Write a program that finds specific stock classes on the file."
760 PRINT
770 PRINT"To help you out, I have made a program that satisfies the "
780 PRINT"requirements of this assignment. As before, you may have it "
790 PRINT"sent to your printer so you can examine it at your leisure."
800 PRINT
810 PRINT"Which do you want:"
820 PRINT
830 PRINT"A  RETURN TO MENU"
840 PRINT"B  LIST PROGRAMS TO PRINTER"
850 PRINT
860 INPUT"Enter your Choice";T$
870 IF T$ = "A" THEN RUN"MENU"
880 IF T$ = "B" THEN PRINT"Be sure to type RUN when the printer stops":LLIST 900-2240
890 GOTO 810
900 REM
910 REM This is the first program.  It requests stock #s and writes 'em to disk
920 REM
```

```
930 REM First we declare our variable as double precision and then open a file
940 REM
950     DEFDBL X
960     INPUT"What is the name of the disk file you want to OPEN";F$
970     OPEN"O",1,F$
980 REM
990 REM Then we ask for stock numbers
1000 REM
1010     INPUT"Please enter a stock number  (-1 to quit)";X
1020     IF X < 0 THEN GOTO 1110
1030 REM
1040 REM  Then put them on disk and go back for more
1050 REM
1060    PRINT#1,X
1070    PRINT
1080    GOTO 1010
1090 REM
1100 REM
1110     END
1120 REM
1130 REM
1140 REM This is the second program.  It reads in the file made by the
1150 REM first program, prints it back to a new file, and then lets you
1160 REM add data to the end of the new file.
1170 REM
1180 REM   First, declare our variable as double precision and open the file
1190 REM
1200     DEFDBL X
1210     INPUT"What is the name of the old data file";F$
1220     OPEN"I",1,F$
1230 REM
1240 REM    Then open the new file (note that there are different buffer #s)
1250 REM
1260     INPUT"What is the name of the new data file you want to create";N$
1270     OPEN"O",2,N$
1280 REM
1290 REM Now read in the data from the old file and print it out to the new
1300 REM file until the old file is empty (see lesson 4 part 2, EOF)
1310 REM
1320     IF EOF(1) THEN GOTO 1390
1330     INPUT#1,X
1340     PRINT#2,X
1350     GOTO 1320
1360 REM
1370 REM When the old file is empty, the program jumps to the CLOSE1 statement
1380 REM
```

```
1390     CLOSE 1
1400     PRINT
1410     PRINT"The file is transferred. You can start adding new data now"
1420     PRINT
1430     INPUT"Enter a stock number (-1 to quit)";X
1440     IF X < 0 THEN GOTO 1470
1450     PRINT#2,X
1460     GOTO 1430
1470 REM
1480 REM Now we close the file, offer to KILL the old file, and end
1490 REM
1500     CLOSE
1510 REM
1520     INPUT"Do you want to KILL the old file (Y/N)";T$
1530     IF T$ = "Y" THEN KILL F$
1540     END
1550 REM
1560 REM
1570 REM  This is the third program.  It searches a disk file and tells you
1580 REM  the number of occurances of a stock class
1590 REM
1600 REM  First, declare our variable as double precision and open a file
1610 REM
1620     DEFDBL X
1630 REM
1640      INPUT"What is the file of stock numbers that you want to search";F$
1650      OPEN"I",1,F$
1660 REM
1670 REM  I've included an option to search for ALL the stocknumbers
1680 REM
1690      INPUT"What is the stock class you are looking for (1 = ALL)";N
1700      PRINT
1710      PRINT"Here are the stock numbers in that class"
1720      PRINT
1730      IF EOF(1) THEN GOTO 1820
1740      INPUT#1,X
1750      IF N=1 THEN PRINT X
1760      IF N=1 THEN GOTO 1730
1770      IF INT(.0000001#X)=N THEN PRINT X
1780      GOTO 1730
1790 REM
1800 REM Once all the data has been checked, we close the file
1810 REM
1820     CLOSE
1830     PRINT
1840     INPUT"End of file.  Do again (Y/N)";T$
```

```
1850     IF T$ = "Y" THEN RUN 1590
1860     END
1870 REM
1880 REM
1890 REM  THIS IS THE ALTERNATE WAY TO UPDATE A SEQUENTIAL FILE. IT IS MORE
1900 REM  EFFICIENT THAN THE FIRST METHOD
1910 REM  IN THIS VERSION, WE READ EVERYTHING INTO AN ARRAY, UPDATE THE ARRAY
1920 REM  AND SEND THE ARRAY BACK OUT TO THE ORIGINAL FILE.  FIRST WE DIM
1930 REM  AN ARRAY TO THE AMOUNT NECESSARY TO HOLD THE FILE. THEN READ IT IN
1940 REM
1950     DIM A(1000)
1960     DEFDBL A    : 'THIS DEFINES ANY VARIABLE STARTING WITH A AS DOUBLE PREC.
1970     INPUT"WHAT IS THE NAME OF YOUR INPUT FILE";F$
1980     OPEN"I",1,F$
1990     N = 0    :' N IS A COUNTER FOR THE ARRAY
2000     IF EOF(1) THEN GOTO 2040
2010     N = N+1
2020     INPUT#1,A(N) :'AS N INCREMENTS,ARRAY POCKETS ARE FILLED WITH DISK DATA
2030     GOTO 2000
2040     CLOSE
2050     PRINT
2060     PRINT"YOUR FILE IS READ INTO THE ARRAY, YOU MAY NOW ADD DATA ONTO IT"
2070     PRINT
2080     N = N+1
2090     INPUT"ENTER A STOCK NUMBER, OR -1 TO QUIT";A(N)
2100     IF A(N) < 0 THEN GOTO 2120
2110     GOTO 2080
2120 REM
2130 REM  NOW WE WRITE IT OUT TO DISK (EXCEPT FOR THE -1 THAT WE ENTERED LAST)
2140 REM
2150     OPEN"O",1,F$
2160     FOR Y = 1 TO N-1
2170        PRINT#1,A(Y)
2180     NEXT Y
2190     CLOSE
2200     END
2210 REM
2220 REM   VOILA' WE DID ALL OUR EDITING WITH THE SAME FILE!
2230 REM
2240 REM
2250 REM this subroutine clears the screen
2260 REM
2270 FOR X = 1 TO 24
2280     PRINT
2290 NEXT X
2300 RETURN
```

APPENDIX E

REPORT PROGRAM LISTING

```
***** Listing of Program 'REPORT' *****                    07/12/83 - 01:02:10

 10 REM This program started on 28 June 1983
 20 REM Version number is 1 August 1983
 30 REM
 40 REM Author - Capt Danny J. Creagan, AFIT
 50 REM
 60 REM Purpose - To read student scores and prepare a training
 70 REM           report
 80 REM
 90 REM variables:
100 REM           N$ = array that holds names of students
110 REM           S1 - S6 = arrays that hold scores for tests 1-6
120 REM           I,J,K,L,X,Y,N,T$,T1-T6 = temporary variables & counters
130 REM
140 CLEAR 10000
150 DIM N$(600),S1(600),S2(600),S3(600),S4(600),S5(600),S6(600)
160 ON ERROR GOTO 1680
170 X=0:Y=1
180 REM
190 REM WE READ IN ALL THE SCORES WITH THE FOLLOWING
200 REM ROUTINES
210 REM
220 GOSUB 1740
230 PRINT"                  COMPUTER ASSISTED INSTRUCTION"
240 PRINT"                            IN BASIC"
250 PRINT
260 PRINT:PRINT:PRINT
270 PRINT"THIS PROGRAM READS IN ALL THE SCORES FROM THE DIFFERENT TEST"
280 PRINT"FILES."
290 PRINT
300 PRINT"IF YOU WISH TO CONTINUE WITH IT, BE SURE THE SCORES YOU WANT"
310 PRINT"PRINTED OUT ARE ON ONE OF YOUR ACTIVE DRIVES."
320 PRINT
330 INPUT"DO YOU WISH TO CONTINUE (Y/N)";T$
340 IF LEFT$(T$,1) = "N" THEN GOSUB 1740 : PRINT"GOING TO MENU":RUN"MENU"
350 GOSUB 1740
360 PRINT"PROGRAM STARTING NOW"
370 PRINT
380 OPEN "I",1,"SCORE1"
390 IF Y = 7 THEN GOTO 440
400 IF EOF(1) THEN GOTO 440
410 Y=X+1
420 INPUT#1,N$(X),S1(X)
430 GOTO 400
440 CLOSE: X=0: OPEN"I",1,"SCORE2"
450 IF Y = 7 THEN GOTO 500
460 IF EOF(1) THEN GOTO 500
```

```
470 X=X+1
480 INPUT#1, N$(X),S2(X)
490 GOTO 430
500 CLOSE:Y=3:OPEN"I",1,"SCORE3"
510 IF Y = 7 THEN GOTO 550
520 IF EOF(1) THEN GOTO 550
530 X=X+1
540 INPUT#1, N$(X),S3(X)
550 GOTO 520
560 CLOSE:Y=4:OPEN"I",1,"SCORE4"
570 IF Y = 7 THEN GOTO 620
580 IF EOF(1) THEN GOTO 620
590 X = X+1
600 INPUT#1, N$(X),S4(X)
610 GOTO 580
620 CLOSE:Y=5:OPEN"I",1,"SCORE5"
630 IF Y=7 THEN GOTO 650
640 IF EOF(1) THEN GOTO 680
650 X = X+1
660 INPUT#1, N$(X),S5(X)
670 GOTO 640
680 CLOSE:Y=6:OPEN"I",1,"SCORE6"
690 IF Y=7 THEN GOTO 740
700 IF EOF(1) THEN GOTO 740
710 X=X+1
720 INPUT#1, N$(X),S6(X
730 GOTO 700
740 CLOSE:Y=1
750 REM
760 REM NOW WE SEARCH FOR DUPLICATE NAMES
770 REM
780 PRINT
790 PRINT"Scores are read in, now I'm consolidating names ";
800 FOR N = Y+1 TO X
810  IF N$(N) = CHR$(124) THEN GOTO 830:'If leg already checked, then skip it
820  IF N$(N) = N$(Y) THEN GOSUB 1560 :'If name is duplicate, call subroutine
830 NEXT N
840 PRINT".";
850 Y=Y+1
860 IF Y < X-1 THEN GOTO 800
870 REM
880 REM WE CAN SORT THEM HERE, BUT IT TAKES A WHILE SO
890 REM USER HAS THE OPTION TO GO ON WITHOUT A SORT
900 REM
910 PRINT
920 PRINT"The computer can sort out the names if you wish, however,
```

```
***** Listing of Program 'REPORT' *****                    07/12/83 - 01:02:19

 930 PRINT"BASIC sorts take a few moments.":PRINT:PRINT:INPUT"Do you want to sort the names
     (Y/N)";T$
 940 IF LEFT$(T$,1) = "y" OR LEFT$(T$,1) = "Y" THEN GOSUB 1810
 950 REM
 960 REM THIS PRINTS OUT REPORT TO SCREEN
 970 REM
 980 PRINT
 990 LINE INPUT"What is today's date (month/day/year)? ";T1$
1000 PRINT"Is ";T1$;" right (Y/N)";:INPUT T$
1010 IF LEFT$(T$,1) = "N" THEN GOTO 990
1020 IF LEFT$(T$,1) = "n" THEN GOTO 990
1030 PRINT
1040 INPUT "DO YOU WANT SCORES SENT TO LINE PRINTER (Y,N)";T$
1050 IF LEFT$(T$,1) = "Y" THEN GOTO 1300
1055 GOSUB 1740
1060 PRINT"CAI IN BASIC                              REPORT DATE: ";T1$
1070 PRINT:PRINT
1080 PRINT TAB(3)"NAME"TAB(16)"TEST 1"TAB(21)"TEST 2"TAB(29)"TEST 3"TAB(37)"TEST 4"TAB(45)"TEST
     5"TAB(54)"TEST 6"
1090 PRINT
1100 PRINT
1110 FOR X = 1 TO Y+1
1120  IF N$(X) = CHR$(124) THEN GOTO 1220
1130  IF LEN(N$(X)) = 0 THEN GOTO 1220
1140  PRINT N$(X);
1150  PRINT TAB(15);:IF S1(X) = 0 THEN PRINT"***";ELSE PRINT S1(X);
1160  PRINT TAB(23);:IF S2(X) = 0 THEN PRINT"***";ELSE PRINT S2(X);
1170  PRINT TAB(31);:IF S3(X) = 0 THEN PRINT"***";ELSE PRINT S3(X);
1180  PRINT TAB(39);:IF S4(X) = 0 THEN PRINT"***";ELSE PRINT S4(X);
1190  PRINT TAB(48);:IF S5(X) = 0 THEN PRINT"***";ELSE PRINT S5(X);
1200  PRINT TAB(56);:IF S6(X) = 0 THEN PRINT"***";ELSE PRINT S6(X);
1210  PRINT
1220 NEXT X
1230 PRINT
1240 PRINT
1250 PRINT"END OF PROGRAM - HIT BREAK OR CONTROL C TO QUIT";
1260 GOTO 1260
1270 REM
1280 REM THIS SECTION PRINTS SCORES TO LINEPRINTER
1290 REM
1300 LPRINT:LPRINT:LPRINT"DATE: "T1$:LPRINT:LPRINT:LPRINT:     LPRINT"       TRAINING REPORT FOR
     COMPUTER ASSISTED INSTRUCTION IN BASIC"
1310 LPRINT:LPRINT
1320 LPRINT TAB(3)"NAME"TAB(13)"TEST 1"TAB(23)"TEST 2"TAB(33)"TEST 3"TAB(43)"TEST 4"TAB(53)"TEST
     5"TAB(63)"TEST 6"
1330 LPRINT
```

```
1340 FOR X = 1 TO Y+1
1350   IF N$(X) = CHR$(124) THEN GOTO 1450
1360   IF LEN(N$(X)) = 0 THEN GOTO 1450
1370   LPRINT N$(X);
1380   LPRINT TAB(20);:IF S1(X) = 0 THEN LPRINT "***";ELSE LPRINT S1(X);
1390   LPRINT TAB(30);:IF S2(X) = 0 THEN LPRINT "***";ELSE LPRINT S2(X);
1400   LPRINT TAB(40);:IF S3(X) = 0 THEN LPRINT "***";ELSE LPRINT S3(X);
1410   LPRINT TAB(50);:IF S4(X) = 0 THEN LPRINT "***";ELSE LPRINT S4(X);
1420   LPRINT TAB(60);:IF S5(X) = 0 THEN LPRINT "***";ELSE LPRINT S5(X);
1430   LPRINT TAB(70);:IF S6(X) = 0 THEN LPRINT "***";ELSE LPRINT S6(X);
1440   LPRINT
1450 NEXT X
1460 REM
1470 REM FOLLOWING LINE CAUSES FORMFEED ON MOST PRINTERS
1480 REM YOU MAY HAVE TO CHANGE IT FOR YOUR MACHINE
1490 REM
1500 LPRINT CHR$(140)
1510   PRINT"END OF PROGRAM - PRESS BREAK OR CONTROL C TO END";
1520 GOTO 1520
1530 REM
1540 REM THIS ROUTINE CONSOLIDATES DUPLICATE NAMES
1550 REM
1560 IF S1(N)>S1(Y) THEN S1(Y)=S1(N)
1570 IF S2(N)>S2(Y) THEN S2(Y)=S2(N)
1580 IF S3(N)>S3(Y) THEN S3(Y)=S3(N)
1590 IF S4(N)>S4(Y) THEN S4(Y)=S4(N)
1600 IF S5(N)>S5(Y) THEN S5(Y)=S5(N)
1610 IF S6(N)>S6(Y) THEN S6(Y)=S6(N)
1620 N$(N) = CHR$(124):'replace the duplicate name with a flag
1630 RETURN
1640 REM
1650 REM THIS IS ERROR TRAP - IT PREVENTS PROGRAM STOPPING
1660 REM WHEN ONE OF THE SCORE FILES IS NOT FOUND
1670 REM
1680 PRINT"FILE";Y;"NOT FOUND"
1690 Y=7
1700 RESUME NEXT
1710 REM
1720 REM THIS CLEARS THE SCREEN ON MOST MACHINES
1730 REM
1740 FOR X = 1 TO 24
1750  PRINT
1760 NEXT X
1770 RETURN
1780 REM
1790 REM this is the bubble sort used to put the names in order
```

```
1800 REM
1810 I = X-1
1820 PRINT
1830 PRINT"Sorting through the names now ";
1840 FOR J=1 TO I
1850     K=J+1
1860     FOR L=X TO K STEP -1
1870          IF N$(L)=<N$(J) THEN GOTO 1980
1880          REM
1890          REM save first value
1900          REM
1910          T$=N$(L):T1=S1(L):T2=S2(L):T3=S3(L):T4=S4(L):T5=S5(L):T6=S6(L)
1920          REM
1930          REM swap array segments
1940          REM
1950          N$(L)=N$(J):S1(L)=S1(J):S2(L)=S2(J):S3(L)=S3(J)
1960          S4(L)=S4(J):S5(L)=S5(J):S6(L)=S6(J)
1970          N$(J)=T$:S1(J)=T1:S2(J)=T2:S3(J)=T3:S4(J)=T4:S5(J)=T5:S6(J)=T6
1980     NEXT L
1990     PRINT".";
2000 NEXT J
2010 PRINT
2020 RETURN
```

SELECTED BIBLIOGRAPHY

REFERENCES CITED

1. Burke, Robert L., CAI Sourcebook. Prentice Hall, Englewood Cliffs NJ. 1982.

2. Carew, Elizabeth. Introduction to BASIC. Computer Assisted Instruction package. University of Alaska. June, 1981.

3. Carter, Lieutenant Colonel Bob, USAF, and Nunley, Lieutenant Mike, USAF. "The Micros are Coming." TAC ATTACK, October, 1981. Page 4.

4. Cokerly, Major, USAF Air Staff. Telephone Interview, January, 1983.

5. Coward, Lieutenant James L.,USAF. "Management of Small Computers in Tactical Air Command." TAC Regulation 300-12. May 1982.

6. Daneman, BASIC Programming, QMT 365, Air Force Institute of Technology. 1983.

7. Freedman, Carol R. and others. Author Training Course (for programmed instruction), US Navy Training Command Publication, NAVEDTRA 10003.

8. Goldstein, Larry, and Goldstein, Martin, IBM PC - Introduction to Programming and Applications. Robert J. Brady Co., Bowie MA. 1982.

9. Holtzman, J. and Silvern, M. Techniques of Programmed Instruction. Wiley Press, New York NY. 1970.

10. Kirkpatrick, Colonel, USAF, Tactical Air Command Director of Operations, Letter Dated 8 February 1983. (references phone conversation with Colonel Lyon).

11. Lehman, H. "The Systems Approach to Education." Audiovisual Instruction. February 13, 1968.

12. Lewis, T. G., Using the Osborne I Computer. Reston Publishing, Reston VA. 1982.

13. Lyon, Colonel, USAF, Tactical Air Command Deputy Director of Operations. Telephone Interview. January, 1983.

14. Lysaught, J. P. and Williams, C.M. A Guide to Programmed Instruction. Wiley Press, New York NY. 1963.

15. Meredith, J. C. "The CAI Author Instructor." Educational Technology Publications, Englewood Cliffs NJ. 1971.

16. Stewart, George. Getting Started With TRS-80 BASIC. LeWay Composing Service. Fort Worth TX, 1981.

17. Stuart, Senior Master Sergeant, NCOIC, USAF CDTS Center. Telephone Interview, January, 1983.

18. Strategic Air Command Regulation 300-4

19. Tactical Air Command Regulation 300-12

20. Weber, Captain, OIC, USAF CDTS Center. Telephone Interview, February, 1983.

21. Welch, and others. Commodore Super Pet Computers. Howard Sams Co., Indianapolis IN. 1981