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

1/
COMPUTER ASSISTED INSTRUCTION
IN BASIC

Captain Danny J. Creagan, USAF

LSSR 23-83

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

Wright-Patterson Air Force Base, Ohio
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.
**Title:** Computer Assisted Instruction in BASIC

**Authors:** Danny J. Creagan, Captain, USAF

**Performing Organization:** School of Systems and Logistics, Air Force Institute of Technology, WPAFB OH

**Report Date:** 28 September 1983

**Number of Pages:** 510

**Security Classification:** UNCLASSIFIED

**Distribution Statement:** Approved for public release; distribution unlimited

**Keywords:**
- Computers
- Programmed Instruction
- Computer Aided Instruction
- Programming Languages
- Microcomputers

**Abstract:** Thesis Chairman: Jeffrey C. Daneman, GS-13, AFIT/LSQ
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.
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, SA
Captain, USAF

September 1983

Approved for public release; distribution unlimited
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

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. BACKGROUND</td>
<td>1</td>
</tr>
<tr>
<td>COMPUTER ASSISTED INSTRUCTION IN BASIC</td>
<td>1</td>
</tr>
<tr>
<td>ISSUE</td>
<td>1</td>
</tr>
<tr>
<td>PROBLEM</td>
<td>2</td>
</tr>
<tr>
<td>RESEARCH QUESTION</td>
<td>6</td>
</tr>
<tr>
<td>2. METHODOLOGY</td>
<td>8</td>
</tr>
<tr>
<td>THE CAI TECHNIQUES ON A MICROCOMPUTER</td>
<td>8</td>
</tr>
<tr>
<td>Physical Constraints on Design of CAI</td>
<td>11</td>
</tr>
<tr>
<td>Development of the CAI Draft</td>
<td>13</td>
</tr>
<tr>
<td>Coding of the CAI Program</td>
<td>14</td>
</tr>
<tr>
<td>Validation</td>
<td>14</td>
</tr>
<tr>
<td>3. CAI PROGRAM FORMAT</td>
<td>16</td>
</tr>
<tr>
<td>PROGRAM LIBRARY</td>
<td>16</td>
</tr>
<tr>
<td>Program Purpose</td>
<td>16</td>
</tr>
<tr>
<td>Flowcharts</td>
<td>17</td>
</tr>
<tr>
<td>Lessons</td>
<td>17</td>
</tr>
<tr>
<td>Flowchart for Main Menu</td>
<td>19</td>
</tr>
<tr>
<td>Flowchart for Lessons One Through Six</td>
<td>20</td>
</tr>
<tr>
<td>CHAPTER</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>Tests 1 - 6</td>
<td>21</td>
</tr>
<tr>
<td>Flowchart for Tests One Through Six</td>
<td>22</td>
</tr>
<tr>
<td>Report Program</td>
<td>24</td>
</tr>
<tr>
<td>Index of Subjects</td>
<td>26</td>
</tr>
<tr>
<td>Program Limitations</td>
<td>29</td>
</tr>
<tr>
<td>4. RECOMMENDED AREAS FOR FURTHER RESEARCH</td>
<td>31</td>
</tr>
<tr>
<td>APPENDICES</td>
<td>33</td>
</tr>
<tr>
<td>APPENDIX A: INSTALLATION INSTRUCTIONS FOR THE CAI PROGRAM LIBRARY ON THE TRS-80 AND CROMEMCO MICROCOMPUTERS</td>
<td>34</td>
</tr>
<tr>
<td>APPENDIX B: PROGRAM LISTINGS FOR THE CAI PROGRAM LIBRARY</td>
<td>39</td>
</tr>
<tr>
<td>APPENDIX C: OUTPUT OF THE CAI PROGRAM LIBRARY</td>
<td>259</td>
</tr>
<tr>
<td>APPENDIX D: HOMEWORK MODULES</td>
<td>481</td>
</tr>
<tr>
<td>APPENDIX E: REPORT PROGRAM LISTING</td>
<td>494</td>
</tr>
<tr>
<td>SELECTED BIBLIOGRAPHY</td>
<td>500</td>
</tr>
<tr>
<td>REFERENCES CITED</td>
<td>501</td>
</tr>
</tbody>
</table>
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 DAR (Data Automation Requirement) by the unit (the DAR, in this case, was a formal request which outlined a specific purpose for the computer), and approval of the DAR 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 Cromemco 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 SQ-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 COTS 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 COTS 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 COTS 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 COTS 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 voluntarily 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 280 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

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Lessons 1, 1a, and test 1</td>
<td>Teach Introduction to BASIC and Computer terms</td>
</tr>
<tr>
<td>2. Lessons 2, 2a, and test 2</td>
<td>Teach saving programs to disk, REMARKS, and branching.</td>
</tr>
<tr>
<td>3. Lessons 3, 3a, and test 3</td>
<td>Teach Loops, Arrays, and DIM statements</td>
</tr>
<tr>
<td>4. Lessons 4, 4a, and test 4</td>
<td>Teach Printer commands &amp; Sequential file I/O</td>
</tr>
</tbody>
</table>
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


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

**MENU CHOICES:**
1. Take lesson in fixed order
2. Take lesson according to student wishes
3. Goto next part
4. Run main menu

**LESSON SECTION SUBROUTINE**

**MENU CHOICES:**
1. Review a section
2. Run main menu

**MENU CHOICES**
1. Review a section
2. Run main menu
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

START

INITIALIZE ARRAYS & FLAGS

DISPLAY QUESTIONS

RESPONSE

Yes

CORRECT?

No

GIVE FEEDBACK

DISPLAY CORRECT ANSWER

SET QUESTION FLAGS

(NEXT PAGE)
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
07/31/83

REPORT DATE:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TEST 1</th>
<th>TEST 2</th>
<th>TEST 3</th>
<th>TEST 4</th>
<th>TEST 5</th>
<th>TEST 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROMEMCO</td>
<td>10</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>***</td>
</tr>
<tr>
<td>CURLY</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>D.O.</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>***</td>
<td>***</td>
<td>2</td>
</tr>
<tr>
<td>DAN</td>
<td>***</td>
<td>7</td>
<td>5</td>
<td>***</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>DAN C.</td>
<td>10</td>
<td>7</td>
<td>9</td>
<td>***</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>DAVID</td>
<td>***</td>
<td>***</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>GOLIATH</td>
<td>***</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>LARRY</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>***</td>
</tr>
<tr>
<td>MOE</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>***</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>POSEY</td>
<td>***</td>
<td>***</td>
<td>3</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>TEST</td>
<td>5</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>WIDGET</td>
<td>***</td>
<td>6</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Subject Name</th>
<th>Lesson#/Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Advanced FOR NEXT</td>
<td>2/1</td>
</tr>
<tr>
<td>2. Arrays</td>
<td>3/2</td>
</tr>
<tr>
<td>3. Arrays (intro)</td>
<td>3/2</td>
</tr>
<tr>
<td>4. Branching Introduction</td>
<td>2/2</td>
</tr>
<tr>
<td>5. CLEAR</td>
<td>3/2</td>
</tr>
<tr>
<td>6. CLOSE</td>
<td>4/2</td>
</tr>
<tr>
<td>7. Comparing Variables</td>
<td>1/2</td>
</tr>
<tr>
<td>8. Concatenation</td>
<td>6/1</td>
</tr>
<tr>
<td>9. Counter Variables</td>
<td>2/1</td>
</tr>
<tr>
<td>10. DATA</td>
<td>2/2</td>
</tr>
<tr>
<td>11. DELETE</td>
<td>1/1</td>
</tr>
<tr>
<td>12. DIMension</td>
<td>3/2</td>
</tr>
<tr>
<td>13. EOF</td>
<td>4/2</td>
</tr>
<tr>
<td>Topic</td>
<td>Pages</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>14. Editing Text</td>
<td>6/2</td>
</tr>
<tr>
<td>15. END &amp; STOP</td>
<td>1/1</td>
</tr>
<tr>
<td>16. Filenames</td>
<td>1/2</td>
</tr>
<tr>
<td>17. FOR NEXT</td>
<td>3/1</td>
</tr>
<tr>
<td>18. Functions Overview</td>
<td>5/2</td>
</tr>
<tr>
<td>19. General Information</td>
<td>1/1</td>
</tr>
<tr>
<td>20. GOSUB</td>
<td>5/1</td>
</tr>
<tr>
<td>21. GOTO</td>
<td>2/2</td>
</tr>
<tr>
<td>22. IF</td>
<td>2/2</td>
</tr>
<tr>
<td>23. Immediate Mode</td>
<td>1/1</td>
</tr>
<tr>
<td>24. INPUT</td>
<td>1/2</td>
</tr>
<tr>
<td>25. INPUT#</td>
<td>4/2</td>
</tr>
<tr>
<td>26. I(insert)</td>
<td>6/2</td>
</tr>
<tr>
<td>27. KILL</td>
<td>3/1</td>
</tr>
<tr>
<td>28. Library Functions</td>
<td>5/2</td>
</tr>
<tr>
<td>29. Library Functions (intro)</td>
<td>1/2</td>
</tr>
<tr>
<td>30. LIST</td>
<td>1/1</td>
</tr>
<tr>
<td>31. LLIST</td>
<td>4/1</td>
</tr>
<tr>
<td>32. LOAD</td>
<td>2/1</td>
</tr>
<tr>
<td>33. Loops</td>
<td>3/1</td>
</tr>
<tr>
<td>34. LPRINT</td>
<td>4/1</td>
</tr>
<tr>
<td>35. Nested Subroutines</td>
<td>5/1</td>
</tr>
<tr>
<td>36. NEW</td>
<td>1/1</td>
</tr>
<tr>
<td>37. Numeric Variables (intro)</td>
<td>1/2</td>
</tr>
<tr>
<td>38. ON GOSUB</td>
<td>5/1</td>
</tr>
</tbody>
</table>
39. OPEN ........................................ 4/1
40. PRINT ................................. 1/1
41. PRINT# ............................... 4/2
42. READ ................................. 2/2
43. REMark ................................. 2/1
44. RESTORE ............................... 2/2
45. RETURN ............................... 5/1
46. RUN ..................................... 2/1
47. SAVE ..................................... 2/1
48. Sequential Files ..................... 4/1
49. Software ................................. 1/1
50. Statements and Programs ............. 1/1
51. String Arrays .......................... 6/1
52. String Assignment ..................... 6/1
53. String Functions ........................ 6/1
54. String Variables (intro) ............. 1/2
55. Subroutines ............................ 5/1
56. User Defined Functions ............... 5/2
57. Using Arithmetic in Computers ...... 1/2
58. Variables (general) .................... 1/2
59. nC(hange) ............................... 6/2
60. nD(lete) ............................... 6/2
61. nS(earch) ............................... 6/2
62. nSPACEBAR ............................. 6/2
63. X(tend) ............................... 6/2
**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.

<table>
<thead>
<tr>
<th>AREA NOT INCLUDED IN CAI</th>
<th>APPLICABLE BASIC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Formatted printing</td>
<td>PRINT USING</td>
</tr>
<tr>
<td>2. Direct/Random Access</td>
<td>FIELD, GET, PUT, LOF</td>
</tr>
<tr>
<td></td>
<td>OPEN&quot;R&quot;,1,&quot;1fn&quot;</td>
</tr>
<tr>
<td></td>
<td>LSET, RSET</td>
</tr>
<tr>
<td>3. Graphics</td>
<td>POINT, SET, RESET</td>
</tr>
<tr>
<td></td>
<td>(TRS-80)</td>
</tr>
<tr>
<td>4. ASCII code</td>
<td>SAVE &quot;1fn&quot;,A</td>
</tr>
<tr>
<td>5. Tape input/output</td>
<td>CSAVE, CLOAD (TRS-80)</td>
</tr>
<tr>
<td>6. Error trapping</td>
<td>ON ERROR GOTO, ON ERROR</td>
</tr>
<tr>
<td></td>
<td>GOSUB, RESUME, ERR, ERL</td>
</tr>
<tr>
<td>7. Defining precision</td>
<td>DEFDBL, DEFSNG,</td>
</tr>
<tr>
<td></td>
<td>DEFINT, CVI, CVS CVO,</td>
</tr>
<tr>
<td></td>
<td>CDDBL, CSNG, CINT</td>
</tr>
<tr>
<td>8. Merging programs</td>
<td>MERGE</td>
</tr>
<tr>
<td>9. Screen clearing</td>
<td>CLS</td>
</tr>
</tbody>
</table>
10. Line renumbering ........... NAME (TRS-80), RENUM (C/PM)

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

12. Internal Clock ............... TIMES, 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 warrant 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 and 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.
APPENDIX A

INSTALLATION INSTRUCTIONS FOR THE CAI
PROGRAM LIBRARY ON THE TRS-80 AND
CROMEMCO MICROCOMPUTERS
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 AF3, NY, after October, 1993, and I will supply you with one.

To run the program, turn on the TRS-80, put TRSOOS 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 instructions 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 COOS 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 COOSCOPY 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 COOS, then the disk may be treated as an ordinary data disk.

If your BASIC is set up with default extensions for BASIC (i.e., 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

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MENU</td>
<td>41</td>
</tr>
<tr>
<td>LESSON ONE part 1</td>
<td>43</td>
</tr>
<tr>
<td>LESSON ONE part 2</td>
<td>64</td>
</tr>
<tr>
<td>TEST ONE</td>
<td>78</td>
</tr>
<tr>
<td>LESSON TWO part 1</td>
<td>86</td>
</tr>
<tr>
<td>LESSON TWO part 2</td>
<td>103</td>
</tr>
<tr>
<td>TEST TWO</td>
<td>119</td>
</tr>
<tr>
<td>LESSON THREE part 1</td>
<td>127</td>
</tr>
<tr>
<td>LESSON THREE part 2</td>
<td>141</td>
</tr>
<tr>
<td>TEST THREE</td>
<td>152</td>
</tr>
<tr>
<td>LESSON FOUR part 1</td>
<td>159</td>
</tr>
<tr>
<td>LESSON FOUR part 2</td>
<td>174</td>
</tr>
<tr>
<td>TEST FOUR</td>
<td>185</td>
</tr>
<tr>
<td>LESSON FIVE part 1</td>
<td>193</td>
</tr>
<tr>
<td>LESSON FIVE part 2</td>
<td>203</td>
</tr>
<tr>
<td>TEST FIVE</td>
<td>216</td>
</tr>
<tr>
<td>LESSON SIX part 1</td>
<td>224</td>
</tr>
<tr>
<td>LESSON SIX part 2</td>
<td>238</td>
</tr>
<tr>
<td>TEST SIX</td>
<td>252</td>
</tr>
</tbody>
</table>
10 REM ** THIS PROGRAM STARTED ON 27 MARCH 1983
15 REM ** MODIFIED FOR DODS 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 616-341-800. MR. D. G. C.
270 PRINT#: IN THE AIRCRAFT MAINTENANCE AREA AFTER 1700.
280 PRINT:  
290 INPUT#: PRESS ENTER TO CONTINUE:*TS
300 PRINT#: BY GOSUB 520
310 PRINT#: MENU CHOOSe:
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*
430 PRINT:  
440 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"PRESS1"
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 I = 1 TO 24
530 PRINT"*"
540 NEXT I
550 RETURN
**** Listing of Program 'LESSONI' ****

1000 REM ** THIS PROGRAM STARTED ON 27 MARCH 1983
1010 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
1020 REM ** TITLE: LESSON IA
1030 REM **
1040 REM **
1050 REM **
1060 GOSUB 9970
1070 PRINT 'LESSON: BASIC IA 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 PRINT 'press the ENTER key to continue';T$:
1230 GOSUB 9970:
1240 GOSUB 10040
1250 PRINT 'A I'm taking this part in its entirety.'
1260 PRINT 'B I wish to review selected areas.'
1270 PRINT 'C I want to go to the second part.'
1280 PRINT 'D I want to return to the Menu.'
1290 PRINT
1300 PRINT 'Press either capital A, B, C, or D and then press ENTER';T$:
1310 IF T$ = "D" GOTO 10160
1320 IF T$ = "A" GOTO 10170
1330 IF T$ = "B" GOTO 1450
1340 IF T$ = "C" GOTO 1250
1350 GOSUB 1560
1360 GOSUB 1920
1370 GOSUB 2770
1380 GOSUB 3750
1390 GOSUB 4860
1400 GOSUB 5550
1410 GOSUB 7670
1420 GOSUB 8500
1430 PRINT 'GOING TO SECOND PART - PLEASE STANDBY'
1440 GOTO 10170
1450 GOSUB 9970
1450 GOSUB 10040
1470 PRINT
1490 PRINT "Please type in the number beside the area you wish"
1510 GOTO 1470
1530 IF N = 0 GOTO 10160
1550 GOTO 1450
1570 PRINT
1590 GOSUB 9970
1610 PRINT
1630 PRINT "Throughout all your lessons, you should have your BASIC manual"
1650 PRINT "handy. If you find yourself stumped by a question, you should"
1670 PRINT "look up the answer in the book. If you can’t find it after"
1690 PRINT "an honest attempt, then make a guess and then go on. You will"
1710 PRINT "have an opportunity to review each section again."
1730 PRINT
1750 PRINT "Beginning with this lesson, you will have homework assigned"
1770 PRINT "at the end of each test. If you do the homework, you will"
1790 PRINT "learn more, and, with the techniques you learn, you will find"
1810 PRINT "that you can tackle small programming jobs as soon as you"
1830 PRINT "complete the course."
1850 PRINT
1870 INPUT "Press ENTER"; T1: GOSUB 9970
1890 PRINT
1910 PRINT
1930 PRINT "Throughout the next six lessons, you will be learning"
1950 PRINT "about computers and what they do. Although the course is"
1970 PRINT "titled 'Computer Assisted Instruction in BASIC', you will"
1990 PRINT "also need to learn the terminology of computers, not just"
2010 PRINT "the BASIC programming language. This first lesson will"
2030 PRINT "start with some fundamental ideas, and expand them as"
2050 PRINT "we go along."
2070 PRINT
2090 INPUT "Press ENTER"; T1: GOSUB 9970
2110 PRINT
2130 PRINT "We use computers to process DATA and give us answers to our"
2150 PRINT "problems. To process this DATA, we must communicate with the"
2170 PRINT "computer using two basic computer components. Those comp-
2190 PRINT "are called: HARDWARE AND SOFTWARE."
2210 PRINT
2230 INPUT "Press ENTER"; T1: GOSUB 9970
2250 IF T1 = "2" GOTO 10160
2270 RETURN
Listing of Program 'LESON1'

1720 GOSUB 9970
1730 PRINT "HARDWARE"
1740 PRINT
1750 PRINT "Hardware is the term used to describe the electrical and"
1760 PRINT "mechanical aspects of a computer. Hardware includes the"
1770 PRINT "parts you can physically touch on, or in, your computer."
1780 PRINT
1790 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":$1
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":$1
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 "Wherever 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":$1
**Listing of Program 'LESSON1'**

```plaintext
07/10/83 - 01:03:41

2280 GOSUB 9970
2290 PRINT "Here is a little quiz - answer in capital letters" 
2300 PRINT "and do not include extra spaces or words"
2310 PRINT
2320 PRINT "What component (HARDWARE or SOFTWARE) is mainly" 
2330 PRINT "used to transform your inputs into a form the CPU can" 
2340 INPUT "understand" ; T$
2350 PRINT
2360 IF T$ = "SOFTWARE" THEN PRINT "WRONG - the correct answer is SOFTWARE" 
2370 IF T$ = "SOFTWARE" THEN PRINT "CORRECT"
2380 PRINT
2390 INPUT "press ENTER to continue" ; T$
2400 GOSUB 9970
2410 PRINT "What do the letters CPU stand for" ; T$
2420 PRINT
2430 IF T$ = "CENTRAL PROCESSING UNIT" GOTO 2560
2440 PRINT "WRONG - the correct answer is CENTRAL PROCESSING UNIT"
2450 GOTO 2570
2460 PRINT "CORRECT - now we are learning something!"
2470 PRINT
2480 GOSUB 9970
2490 PRINT "Which of the following can be considered an OUTPUT device" ; T$
2500 PRINT "A Terminals" 
2510 PRINT "B Tapes" 
2520 PRINT "C Disks" 
2530 PRINT "D ALL of the above" 
2540 PRINT
2550 INPUT "which letter do you select" ; T$
2560 IF LEN(T$) = 1 GOTO 2570
2570 PRINT
2580 IF T$ = "D" GOTO 2670
2590 PRINT "WRONG - D ALL of the above is the correct answer"
2600 GOTO 2670
2610 PRINT "CORRECT"
2620 PRINT
2630 INPUT "press ENTER to continue" ; T$
2640 GOSUB 9970
2650 IF T$ = "SOFTWARE" THEN PRINT "SOFTWARE"
2660 PRINT "SOFTWARE is a collection of written rules that control the computer. Software can be divided into two" 
```

---

**Explanation**

This program is designed to quiz the user on basic computer concepts. It begins by introducing a little quiz and instructing the user to answer in capital letters without extra spaces or words. The first question asks which component is mainly used to transform inputs into a form the CPU can understand. The user is given two options: HARDWARE or SOFTWARE. If the user selects HARDWARE, they receive a correct answer message. If they select SOFTWARE, they receive an incorrect answer message. The program then moves on to the next question, asking what the letters CPU stand for. The user is given four options: A Terminals, B Tapes, C Disks, or D ALL of the above. If the user selects D, they receive the correct answer message. If they select any other option, they receive an incorrect answer message. The program then moves on to the next question, asking which of the following can be considered an OUTPUT device. The user is given four options: Terminals, Tapes, Disks, or ALL of the above. If the user selects D, they receive the correct answer message. If they select any other option, they receive an incorrect answer message. The program concludes by reminding the user that software is a collection of written rules that control the computer.
Listing of Program 'LESSON' 

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 super-vises 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!"
**Listing of Program "LESSON1"**

3300 PRINT "Remember, use only capital letters and don't add unnecessary spaces or words."
3310 PRINT "Press ENTER to continue":T$ 3320 PRINT
3330 INPUT "Which letter":T$ 3340 IF LEN(T$) > 1 GOTO 3300
3350 PRINT
3360 IF T$ = "A" GOTO 3380
3370 PRINT "WRONG - the correct answer is A (Hardware)." 3380 PRINT "CORRECT." 3390 PRINT
3500 INPUT "Press ENTER to continue":T$ 3510 PRINT
3520 PRINT "An operating system translates user-written code into a" 3530 PRINT "form that the operator can understand. (True or False)"
3540 PRINT "Which letter":T$ 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 A (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 PRINT "Which letter":T$ 3690 PRINT "You are RIGHT"
700: PRINT
710: INPUT "Press ENTER to continue":T$ 720: GOSUB 12310
730: IF "a" = "B" GOTO 1790 740: RETURN 750: GOSUB 49
Listing of Program 'LESSON'  

3750 PRINT"General Information"
3770 PRINT
3790 PRINT"There have been many programming languages developed over the years. Many were designed to solve specific problems and therefore required a good deal of previous knowledge about computers."
3810 PRINT
3820 PRINT
3830 PRINT"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."
3860 PRINT
3870 PRINT"BASIC was developed at Dartmouth College for use by students who were unfamiliar with computers and needed a language related to everyday speech."
3900 PRINT
3910 INPUT"Press ENTER to continue";T$
3920 GOSUB 9770
3930 PRINT"General Information (cont)"
3940 PRINT
3950 PRINT"BASIC is easier to master than most other languages, because its instructions are very similar to English grammar."
3970 PRINT
3980 PRINT"However, BASIC is not English. A computer must be instructed in precise terms, with no ambiguity. English has many synonymous and imprecise terms."
4000 PRINT
4010 PRINT
4020 PRINT
4030 PRINT
4040 PRINT
4050 INPUT"Press ENTER for more";T$
4060 GOSUB 9770
4070 PRINT"General Information (cont)"
4080 PRINT
4090 PRINT"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:"
4100 PRINT
4110 PRINT"Add 10, 20, 30, 40, and 50. Divide by 5."
4120 PRINT
4130 PRINT"Write the quotient as the answer."
4140 PRINT
4150 PRINT
4160 PRINT"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."
4170 PRINT
4180 PRINT"This average can be stated in one instruction called PRINT."
**** Listing of Program 'LESSON1' ****

4200 INPUT"press ENTER for an example of the PRINT instruction:	"
4201 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 (15+30+24+100+66)/5"
4310 PRINT
4320 PRINT"In this example, the BASIC verb PRINT tells the computer:	"
4330 PRINT"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:	"
4400 GOSUB 9970
4440 PRINT "BASIC is used by nearly every micro and mainframe computer.	"
4450 PRINT
4460 PRINT"There are many "dialects" of BASIC and they are not all:"
4470 PRINT"compatible with each other. For instance, a BASIC program:
4480 PRINT"written in Honeywell BASIC or Applesoft BASIC will not"
4490 PRINT"run on your computer unless it is modified."
4500 PRINT
4501 INPUT"press ENTER:	"
4502 GOSUB 9970
4506 PRINT "The name for the BASIC on your computer is Microsoft BASIC."
4510 PRINT"Microsoft BASIC is supported by most microcomputers than:
4520 PRINT"any other dialect. It is extremely powerful and matches"
4530 PRINT"the computing capability of most other languages. It:
4540 PRINT"is slower than some, but the slowness is relative. Most:"
4550 PRINT"calculations take milliseconds."  
4560 PRINT
4560 PRINT
4570 INPUT"press ENTER:	"
4580 GOSUB 9970
4584 PRINT "Why is BASIC a good general programming language to learn?"
4590 PRINT
4600 PRINT "It can be used by most students and programmers.	"
4610 PRINT "you don't need to know a lot about computers to use it."
4620 PRINT "It is available on most computers."
4630 PRINT "ALL of the above"  
4640 PRINT
4650 PRINT "Press the letter corresponding to the correct answer:"
4651 PRINT "Be sure to enter only capital letters"
4652 PRINT
**** Listing of Program 'LESSON' ****

4200 INPUT"What is your selection?"T9
4201 PRINT
4202 IF T9 = "D" GOTO 4740
4210 PRINT"WRONG - the correct answer is D 'ALL of the above'
4220 PRINT
4230 GOTO 4750
4240 PRINT"CORRECT"
4250 PRINT
4270 INPUT"Press ENTER to continue"T9
4270 GOSUB 1210
4290 IF T9 = "E" GOTO 4750
4290 RETURN
4300 GOSUB 3770
4310 PRINT" STATEMENTS and PROGRAMS"
4320 PRINT
4330 PRINT"The instruction that we saw in the previous example is a"
4340 PRINT"one line command to the computer. When we combine several"
4350 PRINT"statements, we get a more useful COMPUTER PROGRAM."
4360 PRINT
4370 PRINT"The COMPUTER PROGRAM acts as a series of directions for"
4380 PRINT"the machine to follow."
4390 PRINT
4400 PRINT"The statements that make up the program are expressed as"
4410 PRINT"BASIC verbs which denote actions to be taken - THEN"
4420 PRINT"appear sequentially on NUMBERED PROGRAM LINES. Usually:
4430 PRINT"along with the data that is to be acted upon."
4440 PRINT
4450 INPUT"Press ENTER to continue with STATEMENTS & PROGRAMS"T9
4460 GOSUB 9970
4470 PRINT
4490 PRINT"Each BASIC statement consists of a specific arrangement of"
4490 PRINT"elements. These elements are shown below, in the order:
5000 PRINT"The "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" in the statement."
5110 PRINT
5120 INPUT"Press ENTER to continue"T9
5130 RETURN
**Listing of Program 'LESSON'**

```
150 PRINT* STATEMENTS and PROGRAMS (cont.)*
160 PRINT*  C10. PRINT'S statement: must have a line number and these numbers range
170 PRINT*  1 to 888 in most microcomputers that support
180 PRINT*  'IBM BASIC. Microsoft is the company that owns
190 PRINT*  'the copyright to the particular dialect of BASIC that
200 PRINT*  'is in this machine. It is advisable to write progr
210 PRINT*  'lines in increments of 10 to allow you to insert additional
220 PRINT*  'lines without having to remember every statement line.
230 PRINT* 'The statements are executed in ascending numerical order.
240 PRINT* 'not in the order they were entered.'
250 PRINT*   520 PRINT 'Look up the RENUM. or NAME command in your manual for more.'
260 PRINT* Informationen.'
270 PRINT*  700 PRINT* INPUT press ENTER to continue: '*
280 GOSUB 9999
290 PRINT* STATEMENTS and PROGRAMS (cont.)*
300 PRINT*  C80. PRINT'Do you get the program to execute you use the RUN command.'
310 PRINT*  560 PRINT 'Now for a few questions to see how you are doing.'
320 PRINT*  580 PRINT* INPUT press ENTER for the questions: '*
330 GOSUB 9999
340 PRINT* 'A computer program is a series of *************
350 PRINT*  550 PRINT 'Verbs'
360 PRINT*  540 PRINT 'Words'
370 PRINT 'Statements'
380 PRINT* Synonyms'
390 PRINT*  550 PRINT* INPUT 'Be in the latter opposite the correct answer and press ENTER: '*
400 PRINT*  560 IF 's = 'T' GOTO 550
410 PRINT*  570 GOSUB 560
420 PRINT*  580 PRINT 'The correct answer is: STATEMENTS.'
430 GOTO 5550
440 PRINT*  590 PRINT* CORRECT'
450 PRINT*  600 PRINT* INPUT press ENTER to continue: '*
460 GOSUB 9999
```
5300 PRINT "Which of the following is incorrect?"
5310 PRINT
5320 PRINT "A: A BASIC word is a word that a BASIC processor understands"
5330 PRINT "B: A statement can have no more than two line numbers"
5340 PRINT "C: Data are the recipients of the action of BASIC verbs"
5350 PRINT "D: Line Numbers are written sequentially."
5360 PRINT
5370 PRINT "Press the letter that is beside the correct answer."
5380 PRINT "and then press ENTER."
5390 PRINT
5400 INPUT "What is your choice?:"$'
5410 PRINT
5420 IF T = "B" GOTO 5730
5430 PRINT "WRONG - B Only one line number is allowed per statement."
5440 GOTO 5760
5450 PRINT "CORRECT - way to go!"
5460 PRINT
5470 PRINT "Input "press ENTER for the next question:"$
5480 GOSUB 9770
5490 PRINT "Is 59000 a valid statement number inTurbo BASIC?"
5500 PRINT
5510 PRINT "No."
5520 PRINT "Yes."
5530 PRINT
5540 INPUT "press the letter beside the correct answer and then press ENTER:"$
5550 PRINT
5560 IF T = "A" GOTO 5890
5570 PRINT "WRONG - 59000 is too big. Remember, you can only go to 6000."
5580 GOTO 5900
5590 PRINT "CORRECT."
5600 PRINT
5610 INPUT "press ENTER:"$
5620 GOSUB 9710
5630 IF T = "B" GOTO 4800
5640 RETURN
5650 GOSUB 9770
5660 PRINT "" PRINT STATEMENT"
5670 PRINT
5680 PRINT "The BASIC word PRINT is a command that tells the computer..."
5690 PRINT "to output the data that follows to the computer terminal."
5700 PRINT
5710 PRINT "This data can be numbers, variables, or strings."
5720 PRINT "Strings are combinations of words or numbers that are..."
5730 PRINT "to be printed without having any calculations done to them."
5740 PRINT
5750 PRINT
**** Listing of Program 'LESSON' ****

5800 PRINT
5890 PRINT
5891 INPUT 'press ENTER' : T$ 
5890 GOSUB 9970
5900 PRINT "You can control the output caused by the print statement" 
5910 PRINT "in two ways. If you just want what you ENTER printed" 
5912 PRINT "without any calculations done to it, then you enclose the" 
5910 PRINT "data after the PRINT command in quotation marks." 
5920 PRINT 
5923 PRINT "For Example:" 
5926 PRINT 
5927 PRINT : PRINT "CHR$(13)" "Go For It" "CHR$(13)," 
5920 PRINT "20 EN": 
5923 PRINT "RUN" 
5926 PRINT 
5929 PRINT "In BASIC, if you type this in as shown, you get this result:" 
5930 PRINT 
5932 PRINT "Go For It" 
5934 PRINT 
5936 INPUT 'press ENTER' : T$ 
5930 GOSUB 9970 
5940 PRINT "Another example would be:" 
5940 PRINT 
5941 PRINT : PRINT "CHR$(13)" "This is easy" "CHR$(13)," 
5940 PRINT "20 EN": 
5943 PRINT "RUN" 
5946 PRINT 
5949 PRINT "Which would result in:" 
5950 PRINT 
5952 PRINT "This is easy" 
5954 PRINT 
5956 PRINT 
5958 PRINT "Notice that nothing is changed by the computer, the words" 
5960 PRINT "that were commanded to be output were printed exactly as shown." 
5963 PRINT 
5964 PRINT "Enter the second example:" 
5966 INPUT 'press ENTER' for the second example' : T$ 
5960 GOSUB 9970 
5970 PRINT "The second way the print statement is used to control output' 
5972 PRINT "is by NOT enclosing the data in quotation marks. The data is" 
5974 PRINT "then read by the computer and the computer tries to evaluate" 
5976 PRINT "what the data means in mathematical terms. If you have entered" 
5978 PRINT "data that cannot be mathematically manipulated, then you" 
5980 PRINT "get an ERROR message." 

54
***** Listing of Program 'LESSON1' *****

07/10/83 - 01:02:41

520 FOR I = 1 TO 7
530 PRINT
540 NEXT I
550 INPUT "press ENTER to continue"
560 GOSUB 9970
570 PRINT
580 PRINT "Here is an example of data in quotes, and data not in quotes"
590 PRINT "and what the output would look like:"
600 PRINT
610 PRINT
620 PRINT "CHR(34)*I+""CHR(34)"
630 PRINT "CHR(34)"I+"CHR(34)"
640 PRINT "CHR(34)"I+"CHR(34)"
650 PRINT "CHR(34)"I+"CHR(34)"
660 PRINT "CHR(34)"I+"CHR(34)"
670 PRINT "CHR(34)"I+"CHR(34)"
680 PRINT "CHR(34)"I+"CHR(34)"
690 PRINT "CHR(34)"I+"CHR(34)"
700 PRINT "CHR(34)"I+"CHR(34)"
710 PRINT
720 INPUT "press ENTER to continue"
730 GOSUB 9970
740 PRINT*"PRINT (cont)"
750 PRINT
760 PRINT*"PRINT (cont)"
770 PRINT
780 PRINT
790 PRINT*"PRINT (cont)"
800 PRINT
810 PRINT
820 PRINT
830 PRINT
840 PRINT
850 PRINT
860 PRINT
870 PRINT
880 PRINT
890 INPUT "press ENTER to continue"
900 GOSUB 9970
910 PRINT*"PRINT (cont)"
920 PRINT
930 PRINT
940 PRINT
950 PRINT
960 PRINT
970 PRINT
980 PRINT
990 PRINT
100 PRINT
101 PRINT
***** Listing of Program 'LESSON1' *****

1990 PRINT
1990 PRINT "Results in:"  
2000 PRINT
2010 PRINT"A",S+2,"B",S+1  
2020 PRINT
2030 PRINT
2040 PRINT
2050 INPUT
"press ENTER":T$  
2060 GOSUB 9970  
2070 PRINT"A",S+2,"B",S+1  
2080 PRINT
2090 PRINT"Notice how the commas have caused the terms of the"  
2090 PRINT"statement to be spaced across the screen. The spaces"  
2100 PRINT"are similar to TABS on a typewriter, however, the comma reacts"  
2110 PRINT"differently on different terminals. See your BASIC manual"  
2120 PRINT"or ask your system operator how they react on your machine."  
2130 PRINT
2140 PRINT"Commas usually cause 9 spaces between terms!"  
2150 PRINT
2160 PRINT
2170 INPUT  
2180 GOSUB 9970  
2190 PRINT"PRINT (cont)"
2190 PRINT
2200 PRINT
2210 PRINT"The PRINT statement also allows you to output blank lines."  
2220 PRINT"You print blank lines by typing in the line number and then"
2220 PRINT"a PRINT statement without an argument. For example:"  
2230 PRINT
2230 PRINT"10 PRINT"  
2230 PRINT"CHR$(134) "Now is the time to skip"CHR$(134)  
2240 PRINT"CHR$(20)  "  
2240 PRINT"CHR$(20) "a line."CHR$(134)  
2250 PRINT"RUN"
2260 PRINT
2270 INPUT "press ENTER for results":T$  
2280 PRINT
2290 PRINT"Now is the time to skip"  
2300 PRINT
2310 PRINT
2320 PRINT"a line"
2330 PRINT
2340 PRINT"CHR$(134) "Now is the time to skip"CHR$(134)  
2350 PRINT
2360 PRINT"Which of the following statements would cause the above output?"
2370 PRINT
2380 PRINT"CHR$(12+1)"CHR$(134)  
2390 PRINT


*** Listing of Program 'LESSON1' ***

740 PRINT" 20 PRINT 12+10"
745 PRINT
7460 PRINT"C 15 PRINT "CHR$(12)*TEN+CHR$(14)"
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$(12)*10+CHR$(14)"
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 9979
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 7750
7740 PRINT"WRONG - the correct answer is: 10 PRINT"
7750 GOTO 7779
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 9979
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 statements"
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 = 4/2"
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 to"
8130 GOSUB 9970
8140 PRINT"If you want to start the program back up from where you"
8150 PRINT"STOPped 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 = 2+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 PRINT
8300 INPUT"hit ENTER for the rest of"  
8310 GOSUB 9970
8320 PRINT"BREAK in 30"
8330 PRINT
8340 PRINT
8350 PRINT"New, by typing in CONT, the program will resume execution"
8360 PRINT"For example:"
##### Listing of Program "LESSON1" ####

```
3260 PRINT
3270 PRINT"CONT"
3280 PRINT"22"
3290 PRINT
3300 PRINT
3410 PRINT"Notice how the last line (which was PRINT X+Y) was executed?"
3420 PRINT"It was just as if the STOP statement had never been there"
3430 PRINT"As you progress in BASIC, you will find many uses for this"
3440 PRINT"statement."
3450 PRINT
3460 INPUT"press ENTER to continue":TS
3470 GOSUB 10210
3480 IF T$ = "B" GOTO 7320
3490 RETURN
3500 GOSUB 9970
3510 PRINT*IMMEDIATE MODE*
3520 PRINT
3530 PRINT"Microsoft BASIC has a mode called IMMEDIATE. . . ."
3540 PRINT"Whenever you have implemented BASIC in your system, usually by"
3550 PRINT"typing in the word BASIC, (consult your BASIC manual or your"
3560 PRINT"system operator for specific directions on your particular"
3570 PRINT"machine) you will be in the IMMEDIATE mode. In this"
3580 PRINT"mode, you may execute many BASIC statements without having to"
3590 PRINT"type in line numbers or the command RUN. For example:"
3600 PRINT
3610 PRINT"PRINT CHR$(34)"JUST WHEN I THOUGHT I HAD THE HANG OF IT"CHR$(34)
3620 PRINT
3630 PRINT"This line will print the statement within the quotes as soon"
3640 PRINT"as the ENTER key is pressed."
3650 PRINT
3660 INPUT"press ENTER to continue":TS
3670 GOSUB 9970
3680 PRINT*IMMEDIATE (cont)*
3690 PRINT
3700 PRINT"Another example would be:"
3710 PRINT
3720 PRINT"PRINT 93+10+40"
3730 PRINT
3740 PRINT"Which would result in:"n
3750 PRINT
3760 PRINT"143"n
3770 PRINT
3780 PRINT"As you can see, the computer will do the calculations just"
3790 PRINT"as if it were commanded to do it in the normal way."
3800 PRINT
3810 PRINT
```

**** Listing of Program 'LESSON1' ****

3820 PRINT
3830 INPUT "press ENTER"; T$
3840 GOSUB 9970
3850 PRINT"IMMEDIATE (cont)"
3860 PRINT
3870 PRINT"The biggest disadvantage of the IMMEDIATE mode is that the data"
3880 PRINT"is not stored in memory, and cannot be repeated again."
3890 PRINT"It is lost after the initial display, whereas the programs we"
3900 PRINT"looked at before can be run over and over again by merely "
3910 PRINT"typing in the word RUN. Also, the IMMEDIATE mode is limited to"
3920 PRINT"one line of statements at a time."
3930 PRINT
3940 PRINT
3950 INPUT "press ENTER"; T$
3960 GOSUB 9970
3970 PRINT"NEW Statement"
3980 PRINT
3990 PRINT"If you want to clean out the temporary memory in BASIC, all"
4000 PRINT"you have to do is type in the word NEW. BUT BE CAREFUL WITH"
4010 PRINT"THIS COMMAND. It will erase any program you have resident."
4020 PRINT
4030 PRINT
4040 PRINT"Let's say you've been practicing the commands you have learned"
4050 PRINT"so far, and you have put in a lot of line numbers and RUN them."
4060 PRINT"But now you want to start over. You can erase the mess with"
4070 PRINT"the command NEW."
4080 PRINT
4090 INPUT "press ENTER"; T$
4100 GOSUB 9970
4110 PRINT"LIST"
4120 PRINT
4130 PRINT"Suppose you don't know what's there and you want to find out?"
4140 PRINT"Just type in the command LIST. LIST will show you everything"
4150 PRINT"That's in temporary memory."
4160 PRINT
4170 INPUT "press ENTER"; T$
4180 GOSUB 9970
4190 PRINT"DELETE"
4200 PRINT
4210 PRINT"Finally, what if you don't want to type in a new program."
4220 PRINT"you just want to delete a line? You can do that by typing in"
4230 PRINT"DELETE 10, or DELETE 20 or DELETE (line number)."
4240 PRINT
4250 PRINT"If you want to delete a range of line numbers, you type in"
4260 PRINT"DELETE (low range-high range). Say you want to delete lines"
4270 PRINT"15 to 35. You would type in DELETE 15-35, and the lines"
**** Listing of Program 'LESSON' ****

280 PRINT"would no longer be in memory"
290 PRINT
300 INPUT"press ENTER":T$
310 GOSUB 9970
320 PRINT"Which of the following statements would be an example of"
330 PRINT"IMMEDIATE mode in Microsoft BASIC"
340 PRINT
350 PRINT"A 10 PRINT CHR$(34)*AT LAST"CHR$(34)
360 PRINT"RUN"
370 PRINT
380 PRINT"B PRINT 25+2-3"
390 PRINT
400 PRINT"C 32+2"
410 PRINT
420 PRINT"D STOP"
430 PRINT
440 INPUT"press the letter beside the correct answer and press ENTER":T$
450 PRINT
460 IF T$ = "B" GOTO 490
470 PRINT"WRONG - the correct answer is B (PRINT 25+2-3)"
480 GOTO 9500
490 PRINT"RIGHT - you've got the right idea about immediate mode"
500 PRINT
510 INPUT"press ENTER to continue":T$
520 GOSUB 9970
530 PRINT"What command will erase everything in temporary memory?"
540 PRINT
550 INPUT"type in the command using capital letters":T$
560 PRINT
570 IF T$ = "NEW" GOTO 9600
580 PRINT"WRONG - the command is NEW"
590 GOTO 9610
600 PRINT"CORRECTO MUNDO - THAT'S RIGHT!"
610 PRINT
620 INPUT"press ENTER":T$
630 GOSUB 9970
640 PRINT"What command will list all the line numbers and statements?"
650 PRINT"that you have placed in temporary memory"
660 PRINT
670 INPUT"type in the command using capital letters":T$
680 PRINT
690 IF T$ = "LIST" GOTO 9720
700 PRINT"WRONG - the correct answer is LIST"
710 GOTO 9730
720 PRINT"RIGHT YOU ARE!"
730 PRINT

****
4fe4e
Listing of Progra. 'LESSON1' #fm,
07/10/83 - 01:03:41

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 9940
9930 RETURN
9940 REM ** This subroutine clears the screen on any terminal
9950 REM **
9970 FOR I = 1 TO 24
9980 PRINT
9990 NEXT I
9990 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 "LESSON1"
10180 REM **
10190 REM ** THIS SUBROUTINE Lets student review sections again

42
**** Listing of Program 'LESSON1' ****

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

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":TS
1330 IF TS = "D" GOTO 7220
1340 IF TS = "C" GOTO 7230
1350 IF TS = "B" GOTO 1430
1360 IF TS = "A" GOTO 1270
1370 GOSUB 1550
1380 GOSUB 2570
1390 GOSUB 3090
1400 GOSUB 4380
1410 GOSUB 5170
1420 GOSUB 6400
1430 GOSUB 6960
1440 GOSUB 7000
1450 PRINT
1460 PRINT "Press either capital A, B, C, or D and then press ENTER":TS
1470 IF TS = "D" GOTO 7220
1480 IF TS = "C" GOTO 7230
1490 IF TS = "B" GOTO 1430
1500 IF TS = "A" GOTO 1270
1510 GOSUB 1550
1520 GOSUB 2570
1530 GOSUB 3090
1540 GOSUB 4380
1550 GOSUB 5170
1560 GOSUB 6400
1570 GOSUB 6960
1580 PRINT "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."
1590 PRINT "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."
1600 PRINT "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."
1610 PRINT "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."
1620 PRINT "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."
1630 PRINT "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."
1640 PRINT "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."
1650 PRINT "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."
1660 PRINT "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."
1670 PRINT "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."
1680 PRINT "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."
1690 PRINT "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."
1700 PRINT "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."

LIBRARY FUNCTIONS

Many mathematical FUNCTIONS such as square root, trigonometric
functions, and logarithms 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 SQRT for square root.
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  
1840 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 "PRINT SQR(2+6)"  
1950 PRINT "RUN"  
1960 PRINT "Which would give you:"  
1970 PRINT  
1980 PRINT  
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 permaneL41y 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$ = "B" THEN PRINT "WRONG - the correct answer is A (Yes)"
2340 GOTO 2350
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 PARENTHESES"
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 2550
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 "IC, 32, 50, 1, etc., are considered CONSTANTS. Can you guess"
2640 PRINT "why? It's 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 "the values that were variable and were for calculation"
2790 PRINT "purposes, is the same way we treat them in the computer world."
2800 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. If 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$ = "S" GOTO 2570  
3070 RETURN  
3080 GOSUB 3900
Listing of Program 'LESSONIA'

3090 PRINT* Numeric Variables*
3100 PRINT
3110 PRINT"In computers we assign values to variables to ease our job:".
3120 PRINT
3130 PRINT"I = 1+2".
3140 PRINT
3150 PRINT"In this case, the value of 3 would be assigned to I 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 I, but only temporarily. Take the following".
3190 PRINT"example:".
3200 PRINT
3210 PRINT"X = 1+2"
3220 PRINT"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 GSUB 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. AI, XZ, T1, Y2, and FF are".
3360 PRINT"legal variables. IA, ZZ, or S1 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 GSUB 5960
3450 PRINT"I = 4"
3460 PRINT"Y = 7"
3470 PRINT"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"

98
Listing of Program 'LESSONIA'

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 = X2"
3720 PRINT
3730 PRINT "In the above statement, X will be assigned the value of "
3740 PRINT "X squared, 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."
3910 PRINT "You do, you may either leave the LET word in or "
3920 PRINT "comment it out, the Microsoft 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 SY = 2"
3990 PRINT "b 20 X = 2.5"
4000 PRINT "c 15 + X2 = 56"
**** Listing of Program 'LESSONIA' ****

4010 PRINT"D 10 23 = %4"
4020 PRINT
4030 INPUT"press the letter opposite the correct answer and press ENTER":T$   
4040 PRINT
4050 IF T$ = "4" SOTO 4120
4060 PRINT"WRONG - the correct answer is A ! LET X = 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 SOTO 4140
4120 PRINT "CORRECT - Good job!"
4130 PRINT
4140 INPUT press ENTER:Ts
4150 SBSUB 5930
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 + B = C"
4200 PRINT"B + C = A + B"
4210 PRINT"D = A - B"
4220 PRINT"B - D = A + B"
4230 PRINT
4240 INPUT"press the letter opposite the correct answer":T$    
4250 PRINT
4260 IF T$ = "B" SOTO 4120
4270 PRINT"WRONG - the correct answer is D = A - B"
4280 PRINT"you may need to review this section if you did not get this"
4290 PRINT"question right."
4300 PRINT
4310 SOTO 4140
4320 PRINT"CORRECT - that was a 'KEY' concept. you're doing good!"
4330 PRINT
4340 INPUT press ENTER:Ts
4350 SBSUB 5930
4360 IF T$ = "B" SOTO 5090
4370 RETURN
4380 SBSUB 5930:
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."
when you do, the computer will know that this is a string and will not try to manipulate it. In addition, everything that
is enclosed in double quotes must be enclosed in string variables.

String variables (cont.)

String variables (cont.)
***** Listing of Program 'LESSON1A' *****

400 PRINT"RUN"
401 PRINT
405 PRINT"This is really awesome, I mean really."
409 PRINT
410 PRINT"Do you see how the computer treated the data?"
414 PRINT"What would be the output of this program?"
418 PRINT
500 PRINT10 ZZ = CHR$(44):"THIS IS A TEST QUESTION"CHR$(44):
511 PRINT"RUN"
512 PRINT
516 PRINT Type in the correct answer exactly as it would be printed"
520 INPUT "s"
524 PRINT
528 IF s = "THIS IS A TEST QUESTION" SOTO 511)
530 PRINT"WRONG - the correct answer is:"
534 PRINT
538 PRINT"THIS IS A TEST QUESTION"
540 SOTO 511)
552 PRINT"CORRECT"
554 PRINT
558 INPUT "Press ENTER to continue"
562 102.3 "42."
564 IF s = "2" SOTO 413)
568 RETURN
572 SOTO 395)"
576 PRINT"Using Arithmetic"
578 PRINT
582 PRINT"BASIC will let you use arithmetic to figure out almost anything!"
584 PRINT"Mathematical task you would want. BASIC uses these symbols to"n
588 PRINT"represent addition, subtraction, multiplication, division and"n
592 PRINT"exponentiation (raising something to a power). Here they are:"n
596 PRINT
600 PRINT** Symbol ** Meaning ** Example"
604 PRINT
612 PRINT + ** addition ** A+B"
620 PRINT - ** subtraction ** A-B"
624 PRINT * ** multiplication ** A*B"
632 PRINT / ** division ** A÷B"
636 PRINT ** exponentiation ** A^2 "A squared"
644 PRINT
648 PRINT"Parentheses ( ) are also used, just as in algebra!"
652 PRINT
656 INPUT"Press ENTER to continue"
658 102.3 øA"
662 PRINT"Using Arithmetic cont."
666 PRINT
**** Listing of Program 'LESSONIA' ****

270 PRINT "Note that a "CHR$(49)" or "CHR$(74)" always must be used for multiplication" 280 PRINT "for you tried to use an A or I. the computer" 290 PRINT "you think you were trying to put in another variable label" 300 PRINT "and would give you an error message. Also, you cannot use" 310 PRINT ""CHR$(55)" to mean 55. if you do, you will get an error." 320 PRINT "if you put two variables together, like A and B, to make AB." 330 PRINT "You are not multipling them, rather, you just created a NEW" 340 PRINT "variable (AB)" 350 PRINT 360 PRINT "So always remember to use the asterisk for multiplication." 370 PRINT 380 PRINT 390 INPUT "Press ENTER for NEXT" 400 GOSUB 780 410 PRINT "Using Arithmetic (cont)" 420 PRINT 430 PRINT "The symbols we just looked at are called ARITHMETIC OPERATIONS."
440 PRINT "and they may be combined in any order in a BASIC statement."
450 PRINT "However, just like mathematics, the computer will treat some"
460 PRINT "symbols with a higher priority than others. For example:" 470 PRINT 480 PRINT "X^2 = (2*2)-(2*2)"
490 PRINT 500 PRINT "In this statement, the computer will scan the line and do" 510 PRINT "all terms within parenthesis first. Then it" 520 PRINT "will scan for exponentiation, before those operations. Then" 530 PRINT "it will scan for multiplication or division and perform those" 540 PRINT "operations as it comes to them, and finally, it will scan" 550 PRINT "for addition OR subtraction and perform those operations."
560 PRINT 570 PRINT 580 INPUT "Press ENTER for NEXT" 590 GOSUB 780 600 PRINT "Using Arithmetic (cont)" 610 PRINT 620 PRINT "The computer always scans from left to right. It will scan"
630 PRINT "once for each category of symbols. The categories" 640 PRINT "are restated below." 650 PRINT 660 PRINT "Category: Priority:" 670 PRINT 680 PRINT "HIGHEST:" 690 PRINT "Next HIGHEST:" 700 PRINT "Next HIGHEST:" 710 PRINT "LOWEST:" 720 PRINT 730 INPUT "Press ENTER for screen's end" 740 GOSUB 780
** Using Arithmetic (cont.) **

5860 PRINT "On the first scan, the computer would do the terms within the" 5870 PRINT "parenthesis. It would first do exponentiation (2^2 is 4)" 5880 PRINT "and then it would do the multiplication, and finally the additi-" 5890 PRINT "tion. The value inside the parenthesis would be set at 10." 5900 PRINT "Then it would do the terms outside the parenthesis in order of" 5910 PRINT "importance. First it would do the division, then it would" 5920 PRINT "do the addition (because it is scanning from left to" 5930 PRINT "right) and the subtraction last. Finally, it would set the" 5940 PRINT "value of X at 4" 5950 PRINT 5960 INPUT "press ENTER":TI 5970 GOSUB 5960 6020 PRINT "Using Arithmetic" 6030 PRINT 6040 GOSUB 5960 6050 PRINT "Parentheses can be used to establish precedence within a" 6060 PRINT "statement. Suppose you want to make sure that the LAST part of" 6070 PRINT "a statement is calculated FIRST. You can use parenthesis." 6080 PRINT "For example:" 6090 PRINT 6100 PRINT "G = (3-2+1)" 6110 PRINT 6120 PRINT 6130 PRINT 6140 PRINT 6150 PRINT 6160 PRINT 6170 PRINT 6180 GOSUB 5960 6190 PRINT "Which of the following statements will assign the value of 10" 6200 PRINT "to the variable M?" 6210 PRINT 6220 PRINT "M = 3*2-5" 6230 PRINT "M = 5*2 -(3+4)" 6240 PRINT "M = 2*3+1-4" 6250 PRINT "M = 3+5-2" 6260 PRINT 6270 INPUT "press the letter opposite the correct answer and press ENTER":TI 6280 PRINT 6290 IF TI = "A" GOTO 6240 6300 PRINT "Wrong - the right answer is M = 3*2-5"
**** Listing of Program 'LESSON1' ****

610 PRINT "M = 5*(I-19)"  
620 PRINT "M = I0 + 10 - I0."
630 GOTO 6750
640 PRINT "CORRECT!"
650 PRINT
660 INPUT 'Press ENTER to continue': T$
670 GOSUB 7120
680 IF T$ = "B" GOTO 5170
690 RETURN
700 GOSUB 6960
710 PRINT "Comparing Variables"
720 PRINT
730 PRINT "BASIC uses symbols to compare values to determine relationships."
740 PRINT "Such as whether one variable is less than, more than, or equal to another variable. We have already used one of these symbols."
750 PRINT "It is called the equal sign (=). When you start programming,"
760 PRINT "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."
770 PRINT
780 INPUT 'Press ENTER for examples': T$
790 GOSUB 6960
800 PRINT "Comparing Variables (cont.)"
810 PRINT
820 PRINT "Symbol Meaning Example"
830 PRINT "=" equal A=B
840 PRINT "< less than A<B
850 PRINT "> greater than A>B
860 PRINT "<" not equal to A<>B
870 PRINT "<=" less than or equal A<=B
880 PRINT ">= greater than or equal A>=B"
890 PRINT "We will discuss these in more detail in a later lesson."
900 PRINT
910 INPUT 'Press ENTER': T$
920 GOSUB 6960
930 PRINT "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."
940 PRINT
950 PRINT "By now, you should understand the following program."
960 PRINT
Listing of Program 'LESSONIA' 

1770 PRINT"10 PRINT"CHR$(44);"The product of 10 times 8 is "CHR$(34);" 
1780 PRINT"20 PRINT 10 * 8" 
1790 PRINT"30 PRINT" 
1800 PRINT"40 PRINT" 
1810 PRINT"50 END" 
1820 PRINT"RUN" 
1830 PRINT 
1840 PRINT"Your homework assignment will require you to write a program" 
1850 PRINT"similar to this. Line 10 prints the string, line 20 prints" 
1860 PRINT"the mathematical calculation. Lines 30 and 40" 
1870 PRINT"print two blank lines. You make the program work by using" 
1880 PRINT"the RUN word after you have entered the statements." 
1890 PRINT"Your actual homework assignment is at the end of the test." 
1900 PRINT 
1910 INPUT'Enter a statement:' 
1920 RUN 
1930 REM ** 
1940 REM ** This subroutine clears the screen on any terminal 
1950 REM ** 
1960 FOR I = 1 TO 24 
1970 PRINT 
1980 NEXT I 
1990 PRINT 
2000 RETURN 
2010 PRINT* 
2020 PRINT"LESSON 18" 
2030 PRINT"This is the second part of a two part lesson" 
2040 PRINT"It is divided into the following sections." 
2050 PRINT 
2060 PRINT"1 Library Functions" 
2070 PRINT"2. Variables (general)" 
2080 PRINT"3. String Variables" 
2090 PRINT"4. Using Arithmetic" 
2100 PRINT"5. Numeric Variables" 
2110 PRINT"6. Comparing Variables" 
2120 PRINT"7. Lesson Summary" 
2130 PRINT* 
2140 PRINT 
2150 INPUT'Which do you want to do?' 
2160 J:GOTO 7200 
2170 PRINT"Continue on" 
2180 PRINT"Review this lesson again" 
2190 PRINT 
2200 INPUT'Enter a statement:' 
2210 IF T$ = "A" AND T$ <> "B" GOTO 7190 
2220 RETURN 
2230 RUN "TEST1" 
2240 RUN "MENU"
***** Listing of Program 'LESSON1' *****

7230 RUN 'LESSON1'
7240 END
**** Listing of Program 'TESTI' ****

07/10/83 - 02:17:03

1000 REM **
1010 REM ** LESSON: TESTI 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$(X) = NAMES ARRAY, USED TO READ IN SEE-
1070 REM ** TENTIAL NAMES, AND TO WRITE OUT
1080 REM **
1090 REM ** S$(X) = SCORES ARRAY - USED TO READ AND
1100 REM ** WRITE SCORES
1110 REM ** Q$(X) = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM ** CORRECT ANSWERS. IF AN ARRAY
1130 REM ** ELEMNT EQUALS 1, THE ANSWER WAS
1140 REM ** CORRECT
1150 REM **
1160 CLEAR 3000
1170 GOSUB 4130
1180 DIM N$(1000)
1190 DIM S$(1000)
1200 DIM Q$(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 "Yes"
1390 PRINT "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."
**** Listing of Program 'TEST' ****

1460 GOTO 1490
1470 PRINT"CORRECT"
1480 Q(I) = 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)"
1630 PRINT"IST will list your program lines, NEW erases your"
1640 PRINT" program, and ADD is not a legal BASIC word."
1650 PRINT" review part 2, FUNCTIONS"
1660 GOTO 1670
1670 PRINT"CORRECT"
1680 Q(I) = 1
1690 PRINT
1700 INPUT"press ENTER to continue";T$
1710 GOSUB 4130
1720 PRINT"Which statement would print the word TEST"
1730 PRINT
1740 PRINT"A PRINT TEST"
1750 PRINT"B PRINT "CHR$(34)"TEST"CHR$(34)"
1760 PRINT"C OUTPUT "CHR$(34)"TEST"CHR$(34)"
1770 PRINT"D PRINT 'TEST'"
1780 PRINT
1790 INPUT"press the letter opposite the correct answer and press ENTER";T$
1800 PRINT
1810 IF T$ = "B" THEN GOTO 1830
1820 PRINT"WRONG - the correct answer is B (PRINT "CHR$(34)"TEST"CHR$(34))"
1830 PRINT"Answer A would treat the word TEST like a variable."
1840 PRINT"Answer C has an illegal BASIC word (output) and"
1850 PRINT"answer B uses the wrong characters for quotes."
1860 PRINT" review part 1, PRINT, and part 2, String Variables"
1870 GOTO 1950
1880 PRINT"CORRECT"
1890 Q(I) = 1
1900 PRINT
1910 INPUT"press ENTER";T$

79
1870 GOSUB 4130
1880 PRINT "What would the following program's output be?"
1890 PRINT
1900 PRINT "10 = SQR(4)"
1910 PRINT "20 PRINT 5*X"
1920 PRINT "RUN"
1930 PRINT
1940 :INPUT "Type in your answer and press ENTER":TS
1950 PRINT
1960 IF TS = "25" THEN GOTO 2000
1970 PRINT "WRONG - the correct answer is 25"
1980 PRINT "line 10 puts the square root of 4 into the"
1990 PRINT "variable X, line 20 causes 5 to be taken to"
2000 PRINT "the power of 2. 5 squared is 25."
2010 PRINT "review part 1, PRINT, and part 2, FUNCTIONS"
2020 GOTO 2000
2030 PRINT "CORRECT"
2040 GOSUB 4130
2050 PRINT "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":TS
2090 PRINT
2100 IF TS = "10 PRINT " THEN GOTO 2140
2110 PRINT "WRONG - the correct answer is 10 PRINT"
2120 PRINT "review part 1, PRINT"
2130 GOTO 2140
2140 PRINT "CORRECT"
2150 GOSUB 4130
2160 PRINT
2170 INPUT "press ENTER":TS
2180 GOSUB 4130
2190 PRINT "The two types of FUNCTIONS are LIBRARY and COMPUTER, TRUE or"
2200 PRINT "FALSE"
2210 PRINT
2220 PRINT "TRUE"
2230 PRINT "FALSE"
2240 PRINT
2250 INPUT "press the letter opposite the correct answer and press ENTER":TS
2260 PRINT
2270 IF TS = "B" THEN GOTO 2230
2280 PRINT "WRONG - the correct answer is B (False) - the two types"
2290 PRINT "of functions are LIBRARY and USER. Review part"
**** Listing of Program 'TEST1' ****
07/10/83 - 02:17:03

2300 PRINT*  2, FUNCTIONS.*
2310 PRINT
2320 GOTO 2350
2330 PRINT*CORRECT*
2340 Q(s) = 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 SQRT(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)"
2490 PRINT* You cannot set a constant (25) equal to a variable"
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*review part 1, PRINT, part 2, FUNCTIONS*
2590 PRINT
2600 INPUT*Type in the letter opposite the INCORRECT 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 R(0) = 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*
**Listing of Program 'TEST1'**

2750 PRINT "B MEMORY"
2760 PRINT "C KEYBOARD"
2770 PRINT
2780 INPUT "(C) 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)"
2820 PRINT "Answers A & B are not peripherals, they are part of the main computer structure."
2830 PRINT "review part 1, General Information"
2840 GOTO 2860
2850 PRINT "CORRECT"
2860 Q(9) = 1
2870 PRINT
2880 GSUB 4130
2890 PRINT "If you had the following program in memory;"
2900 PRINT
2910 PRINT "0 6=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 "(D) 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)"
3070 PRINT "Answer A would delete the whole program, and"
3080 PRINT "answer B prints the sum of 10 and 40. Answer C would cause a syntax error."
3090 PRINT "review part 1, IMMEDIATE, NEW, DELETE"
3100 GOTO 3110
3110 PRINT "CORRECT"
3120 Q(10) = 1
3130 PRINT
3140 INPUT "(E) press ENTER"; T$ 
3150 GSUB 4130
3160 FOR I = 1 TO 10
3170  Y = Y + Q(I)
3180 NEXT I
**** Listing of Program 'TESTt' ****

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: f?"
3370 IF TI = "S" THEN GOTO 3600
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: f?"
3490 OPEN"D:\".\"SCORE1"
3500 X = 0 
3510 IF EOD THEN GOTO 3570 
3520 X = X+1
3530 INPUT"Yf. N: f?"
3540 INPUT"E (f)"
3550 IF Y = "Y" THEN GOTO 3510 
3560 GOTO 3570 
3570 CLOSE 
3580 X = X+1
3590 "f? = Y
3600 S(X) = Y
3610 OPEN"D:\".\"SCORE1"

82
**** Listing of Program 'TEST1' ****

07/10/83 - 02:17:32

3620 FOR M = 1 TO X
3630 PRINT#INS(W)
3640 PRINT#INS(W)
3650 NEXT M
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:
3720 PRINT
3730 PRINT"THE SUM OF 3, 2, AND 22 IS"
3740 PRINT"27"
3750 PRINT
3760 PRINT"Make the second statement ('27') actually calculate the"
3770 PRINT"sum of 3, 2, and 22. - similar to the example at the"
3780 PRINT"end of the last part of lesson 1. Finally, print two"
3790 PRINT"blank lines at the end of the program."
3800 PRINT"Be sure to copy this instruction down before you go on."
3810 GOTO 4170
3819 S(I) = Y
3820 IF EOF(I) THEN CLOSE: GOTO 3890
3829 I = I+1
3830 INPUT#I, W$1(I), S$1(I)
3839 GOTO 3820
3840 OPEN"0", 1, "SCORE1"
3849 FOR M = 1 TO X
3850 PRINT#INS(W)
3860 PRINT#INS(W)
3870 NEXT M
3880 PRINT
3890 PRINT"You may now take LESSON 2. You will be returned to the MENU"
3900 PRINT"from where you may go to LESSON 2 or quit."
3910 GOTO 4170
3919 IF Y=10 THEN RETURN
3920 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:" www.canadaersc.com/
#### Listing of Program "TEST1" ####

07/10/83 - 02:17:03

4060 PRINT
4070 INPUT"press ENTER\:*\:"
4080 GOSUB 4130
4090 RETURN
4100 REM **
4110 REM ** this subroutine clears the screen**
4120 REM **
4130 FOR I = 1 TO 24
4140 PRINT
4150 NEXT I
4160 RETURN
4170 PRINT
4180 INPUT"press ENTER to return to MENU\:*\:"
4190 RUN"MENU"
***** Listing of Program 'LESSON2' *****

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
1100 PRINT\"LESSON: BASIC 2 VERSION: 1 AUGUST 83\"
1190 PRINT
1200 PRINT
1210 PRINT\"TIME REQUIRED TO COMPLETE LESSON: About One Hour\"
1220 PRINT
1230 PRINT\"AUTHOR: Capt. Danny J. Creagan\"
1240 PRINT\"Air Force Institute of Technology\"
1250 PRINT
1260 PRINT\"OBJECTIVE: To teach the student about permanent storage.\"
1270 PRINT\"how to handle data in BASIC, and how to branch to\"
1280 PRINT\"different parts of a program.\"
1290 PRINT
1300 PRINT
1310 PRINT\"press the ENTER key to continue\":1$\n1320 GOSUB 8120
1330 GOSUB 8130
1340 PRINT\"A\" I\'m taking this part in its entirety.\"
1350 PRINT\"B\" I wish to review selected areas.\"
1360 PRINT\"C\" I want to go to the second part.\"
1370 PRINT\"D\" I want to return to the menu.\"
1380 PRINT
1390 INPUT\"Press extr-\" 4 capital A, B, C, or D and then press ENTER\":4$\n1400 IF 4$ = "D" GOTO 1360
1410 IF 4$ = "C" GOSUB 8370
1420 IF 4$ = "B" GOTO 1430
1430 IF 4$ <> "A" GOTO 1260
1440 GOSUB 1540
1450 GOSUB 1950
1460 GOSUB 3870
1470 GOSUB 5000
1480 GOSUB 5670
1490 GOSUB 8120
1500 GOTO 9370
1510 GOSUB 8120
1520 GOSUB 8130
1530 PRINT
Listing of Program 'LESSON2'

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 8360  
1520 ON N GOSUB 1540,1950,3870,5000,5670  
1530 GOTO 4490  
1540 SDBUS 3120  
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 SDBUS 3120  
1580 PRINT "Here is one possible way to complete your homework:"  
1590 PRINT  
1600 PRINT"10 PRINT CHR$(34)"  
1610 PRINT"THE SUM OF 3, 2, AND 22 IS"  
1620 PRINT"CHR$(34)"  
1630 PRINT"3 + 2 + 22"  
1640 PRINT"THE SUM OF 3, 2, AND 22 IS"  
1650 PRINT"27"  
1660 INPUT "You can try this one if you had trouble with yours. Press ENTER" ; T$  
1670 T$ = ""  
1680 SDBUS 3120  
1690 PRINT "Introduction"  
1700 PRINT "In this lesson we will cover some of the most exciting and"  
1710 PRINT "useful commands in the BASIC language. When we left off,"  
1720 PRINT "in lesson 1, we had discussed some of the fundamental commands"  
1730 PRINT "that you must use just to get BASIC started. Now, we will"  
1740 PRINT "discover how to SAVE our programs for future use, how to"  
1750 PRINT "control data input in our program, and how to leave little"  
1760 PRINT "messages in our program so that other programmers can under-"  
1770 PRINT "stand what we are trying to do. Most importantly, we will"  
1780 PRINT "discover how to branch to different parts of a program"  
1790 PRINT "depending on our data manipulation requirements. That way, one"  
1800 PRINT "program can be extremely flexible and do many different things"  
1810 PRINT "of work for us."  
1820 PRINT  
1830 INPUT "Press ENTER to continue" ; T$  
1840 SDBUS 3120  
1850 PRINT "Introduction"  
1860 PRINT  
1870 PRINT "After taking this lesson, we recommend you practice some of the"
LISTING OF PROGRAM 'LESSON 2' 07/10/83

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 try 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
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
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
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"
2320 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
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 "form format EXACTLY."
2460 PRINT
2470 PRINT "TRS-80 CRONEMCO (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"n
2520 PRINT "is that the TRS-80 has a slash, "/", between the filename"n
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
2580 PRINT
2590 PRINT "TR-80 CRONEMCO"
2600 PRINT
2610 PRINT "(filename)/(extension) (filename).(extension)"
2620 PRINT
2630 PRINT "In the two examples, (filename), is an alphabetical character"n
2640 PRINT "string no longer than eight (8) characters. (extension) is "n
2650 PRINT "a file extension name that is also an alphabetical character"n
2660 PRINT "string. The extension must not be longer than three (3):"n
2670 PRINT "characters. The extension is OPTIONAL but, if used, must"n
2680 PRINT "follow the format EXACTLY. NUMBERS may be used in both "n
2690 PRINT "filenames and extensions, but they must NOT be the FIRST letter"n
2700 PRINT
2710 INPUT "press ENTER"; T$
2720 GOSUB 8120
2730 PRINT
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"
Listing of Program 'LESSON2'

2790 PRINT* GOODNESS GOODNESS.BAS
2800 PRINT
2910 PRINT* Notice that the filenames do not have to make sense, just so they mean something to the programmer who made them. It's unwise to name your program something common,
2920 PRINT* like TEST.BAS, because someone else has probably already used it.
2930 PRINT* That name. If they have, you will destroy their program when you save your program to disk. Each program name must be unique.
2940 INPUT* press ENTER*; T$
2950 GOSUB 8120
2960 PRINT* Filenames (cont)
2970 PRINT* SWIMFIN.BAS
2980 PRINT
2990 PRINT* 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.
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 filenames are and what legal filenames look like. In the future you will use them a lot. Be sure you understand.
3070 PRINT* 'stand 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.'
3080 PRINT
3100 INPUT* press ENTER*; T$
3110 GOSUB 8120
3120 INPUT* ENTER a 'T' if using a RS-232, or 'C' if CP/M or CP/MEMCOMM
3130 IF T$ = "T" GOTO 3140
3140 GOSUB 8120
3150 PRINT* 'In the Crossaco system, which of the following would be considered a legal filename.'
3160 PRINT
3170 PRINT* FILE/FIL
3180 PRINT* 'LESSON.ZAP'
3190 PRINT* 'TEST.BAS'
**** Listing of Program 'LESSON2' ****

3250 PRINT" LUNCHTALK"
3250 PRINT
3270 INPUT"press the letter opposite the correct answer and press ENTER":T$
3280 IF T$ = "A" GOTO 3290
3290 PRINT
3300 PRINT"WRONG the correct answer is A (XXX XXX 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 9270
3520 IF T$ = "B" GOTO 1560
3530 RETURN
3540 GOSUB 9120
3550 PRINT"Which of the following filenames is correct?"
3560 PRINT
3570 PRINT"A" XXXXXX.BAS"
3580 PRINT"B" 24:lesson.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 XXXXXX.BAS"
3660 GOTO 3680
3670 PRINT" CORRECT"
3680 PRINT
3690 INPUT"press ENTER":T$
3700 GOSUB 9120.
****** Listing of Program 'LESSON2' ******

771: PRINT: "An extension to a filename is mandatory."
772: PRINT
773: PRINT* YES
774: PRINT* NO
775: PRINT
776: INPUT"Press the letter opposite the correct answer and press ENTER":TS
777: PRINT
778: IF TS = "S" GOTO 3910
779: PRINT"WRONG - the correct answer is S (an extension is not needed)."
780: GOTO 3920
791: PRINT"CORRECT"
792: PRINT
793: INPUT"Press ENTER":TS
794: GOSUB 9270
795: IF TS = "S" GOTO 7950
796: RETURN
797: GOSUB 3120
798: PRINT"Save, LOAD and RUN"
799: PRINT
800: PRINT
801: PRINT"At the beginning of the last section, we said we would discover"
802: PRINT"how to Save our program so we wouldn't have to keep typing".
803: PRINT"then in all the time. Well, this is it. To SAVE your program,".
804: PRINT"let's say you called it 'MYPROG.BAS'. all you do is:".
805: PRINT
806: PRINT 1) Type in the program
807: PRINT 2) Type SAVE "CHAA:CA1MYPROG.BAS"<CR>
808: PRINT 3) Congratulate yourself on a good job!
809: PRINT
810: PRINT"Be sure to notice that the filename is enclosed in quotation marks."
811: PRINT"This is mandatory, if you don't enclose the name in".
812: PRINT"quotes, the command will 'EDIT' it will fail:".
813: PRINT
814: PRINT
815: INPUT"Press ENTER":TS
816: GOSUB 3910
817: PRINT"SAVE, LOAD, and RUN (cont)."
818: PRINT
819: PRINT 1) PRINT"There are other things you should be aware of before you try".
820: PRINT"to SAVE a program. First, there should be enough room on the".
821: PRINT"disk to hold the program. If you are using a COMMAND hard"
822: PRINT", you will probably not have any problems in this area. Ask".
823: PRINT"YOUR SYSTEM OPERATOR for more information. If you are using".
824: PRINT"a floppy disk, then go to the COMMAND mode by:
825: PRINT"using in CMD"<CR>"S"<CR><CR>" and then typing in DIR to see".
826: PRINT"DIR 1. drive 1 is called 0 and drive 2 is called 1, then".
827: PRINT"match the display you will see the free space left on the disk".
828: PRINT"and a 'DIRECTOR' of the rules on the disk. If you have ever".
"LESSON"

I**ST Ep'"R* "LESSON."**

17:0 PRjNT"DO you have enou@h space for almost any program?"
17:1 PRjNT"SEE YOUR OPERATING MANUAL FOR MORE DETAILS."
17:2 PRjNT
17:3 INPUT"press ENTER"$s
17:4 GOSUB 210$3
17:5 PRINT"
17:6 PRINT"
17:7 PRINT"SAVE, RUN, and LOAD (cont)"
17:8 PRINT
17:9 PRINT
17:10 PRINT"Remember, if you have enough space, then just type in this:
17:11 PRINT
17:12 PRINT"SAVE "CHR$(14)*MYPROG.BAS*CHR$(14)" or "CHR$(14)*MYPROG.BAS*CHR$(14)
17:13 PRINT"if using a TRS-80"
17:14 PRINT
17:15 PRINT"Now that we know how to SAVE a program, how do we get it back?"
17:16 PRINT"so we can RUN it again. That's easy. Just type in:
17:17 PRINT
17:18 PRINT"RUN"CHR$(14)*MYPROG.BAS*CHR$(14)
17:19 PRINT
17:20 PRINT
17:21 INPUT"press ENTER"$s
17:22 GOSUB 210$3
17:23 PRINT"
17:24 PRINT
17:25 PRINT"when you type in the command, the computer will load your"
17:26 PRINT"program THAT YOU HAD PREVIOUSLY SAVED, and RUN it."
17:27 PRINT
17:28 PRINT
17:29 PRINT"What if you just saved a piece of a program because you were
17:30 PRINT"tired or expecting to come back at a later date and add to it?"
17:31 PRINT"if you did that, then you wouldn't want to RUN the program."
17:32 PRINT"you would just want to LOAD the program and LIST it to be"
17:33 PRINT"sure it was the right one, then add the line numbers you need."
17:34 PRINT"to complete the program."
17:35 PRINT
17:36 PRINT
17:37 INPUT"press ENTER"$s
17:38 GOSUB 3120$3
17:39 PRINT"
17:40 PRINT"
17:41 PRINT"SAVE, RUN, and LOAD (cont)"
17:42 PRINT
17:43 PRINT"You would LOAD the program using the same format as for SAVING:"
17:44 PRINT"and RUNNING it. That is:
17:45 PRINT
17:46 PRINT
17:47 PRINT"LOAD"CHR$(14)*MYPROG.BAS*CHR$(14)
17:48 PRINT
17:49 PRINT
17:50 PRINT"SURE TO SAVE THE PROGRAM AGAIN AFTER YOU MODIFY IT, BECAUSE:"
17:51 PRINT"ONLY A COPY OF THE OLD VERSION WILL BE ON THE DISK:"
## Listing of Program "LESSEQ2" ##

07/10/83 - 09:11:18

400 GOSUB 9120
410 PRINT "What is the following incorrect command to SAVE a program named FRITI?"
420 PRINT "A. SAVE FRITI"
430 PRINT "B. LOAD FRITI.BAS"
440 PRINT "C. SAVE MYPROG"
450 PRINT *41*L~"cof Program 'LESSON1'##1##."
460 PRINT "##1###"

470 INPUT "Press the letter opposite the correct answer and press ENTER":7$
480 PRINT
490 IF T$ = "A" GOTO 4770
500 PRINT "The correct answer is A (SAVE"CHR$(241)"FRITI"CHR$(241))"
510 GOTO 4790
520 PRINT "CORRECT - FRITI thanks you!"
530 PRINT
540 INPUT "Press ENTER":7$
550 GOSUB 9120
560 PRINT "You have just gotten to BASIC and want to load a program you have been working on. The filename for the program is SQUEEZE."".
570 PRINT "How would you get the program from permanent storage to temporary memory? Which of the following would you type IN:";
580 PRINT
590 PRINT "A. PRINT "LOAD"CHR$(241)"SQUEEZE"CHR$(241)"
600 PRINT "B. LOAD"CHR$(241)"SQUEEZE.BAS"CHR$(241)"
610 PRINT
620 INPUT "Press the letter opposite the correct answer and press ENTER":7$
630 PRINT
640 IF T$ = "B" GOTO 4750
650 PRINT "The correct answer is B (LOAD"CHR$(241)"SQUEEZE"CHR$(241))"
660 GOTO 4790
670 PRINT "CORRECT - give yourself a HUG!";PRINT
680 INPUT "Press ENTER":7$
690 GOSUB 9270
700 IF T$ = "3" GOTO 3870
710 RETURN
720 GOSUB 9120
730 PRINT *"Remark:*";
740 PRINT
750 PRINT "Something we should start early in our programming life is:";
760 PRINT "DOCUMENTATION of how a program runs. You can include state-
770 PRINT "ments within a program THAT WILL NOT BE TOUCHED BY THE COMPUTER-
780 PRINT "s and will add to the clarity of your program. That way, when
790 PRINT "YOU LIST your program (or when another programmer does), you
800 PRINT "can read the reminders left behind and more fully understand"
PRINT "The program.
PRINT "These statements are called REM statements (REMARK statements):"
PRINT
INPUT "Press ENTER":T$  
GOSUB 8120
PRINT "REMARK (cont)*"  
PRINT
PRINT "The format for a REM statement is (LINE #) REM (REMARK):"
PRINT
PRINT "An example is:"  
PRINT
PRINT "This is now a REM statement is made"  
PRINT
PRINT "When the computer sees REM it ignores all data that follows"  
PRINT
PRINT INPUT "Press ENTER":T$  
GOSUB 8120
PRINT
PRINT "Here is an example of REMarks in a program:"  
PRINT
PRINT "R = 2"  
PRINT "REM the variable x in the next line is approx equal to PI"  
PRINT "x = 3.14"  
PRINT "C = x**2"  
PRINT "PRINT C"  
PRINT "SUN"  
PRINT
PRINT "Which would give:"  
PRINT
PRINT "12.55"  
PRINT
PRINT "Notice that the REM statement was not printed"  
PRINT
PRINT INPUT "Press ENTER":T$  
GOSUB 9120
PRINT "CHR$(14)"  "SWEETUMS is my girl."  "CHR$(14)"  
PRINT "CHR$(14)"  "SWEETUMS is my girl."  "CHR$(14)"  
PRINT "CHR$(14)"  "her real name is CANDY"  "CHR$(14)"  
PRINT "CHR$(14)"  "I love her"  "CHR$(14)"  
PRINT "RUN"  
PRINT
PRINT "The above program would print the following: (TRUE OR FALSE)!"  
PRINT
PRINT "SWEETUMS is my girl."  
PRINT "her real name is CANDY:"  
PRINT
**Listing of Program 'LESSON2'**

```
5550 INPUT "TRUE or FALSE, whichever is correct":T$
5560 PRINT
5570 IF T$ = "TRUE" GOTO 5510
5580 PRINT "WARNING - the correct answer is TRUE - REM statements are not"
5590 PRINT "printed!"
5600 GOTO 5520
5610 PRINT "CORRECT - good job!"
5620 PRINT
5630 INPUT "press ENTER":T$
5640 GOSUB 3270
5650 IF T$ = "B" GOTO 5600
5660 RETURN
5670 GOSUB 3120
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 = I0"
5740 PRINT "20"
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 I in line number 10."
5820 PRINT
5830 INPUT "press ENTER":T$
5840 GOSUB 3120
5850 PRINT " INPUT Statements (cont)"
5860 PRINT
5870 PRINT "It is also possible to assign data while the program is run -"
5880 PRINT "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 "INPUT" "CHR$(24)" press the correct letter, then press ENTER" "CHR$(24)"
5950 PRINT
5960 INPUT "press ENTER":T$
5970 GOSUB 3120
5980 PRINT " INPUT Statements (cont)"
5990 PRINT
6000 PRINT "CHR$(24)" press the correct letter, then press ENTER" "CHR$(24)"
```
Listing of Program 'LESSON2' 07/10/83 - 00:01:13

5610 PRINT 'RUN'
5630 PRINT
5630 PRINT 'Gives us: '
5640 PRINT
5650 PRINT 'Press the correct letter, then press ENTER: '
5660 PRINT
5670 PRINT 'Notice that a question mark is automatically inserted after the'
5680 PRINT 'message is printed. When the question is answered, the letter'
5690 PRINT 'that the student selects is assigned to TS, just as if we'
5700 PRINT 'had assigned a value to it in an equals statement: '
5710 PRINT 'Also note that a semi-colon is placed after the text:'
5720 PRINT
5730 INPUT 'Press ENTER: TS'
5740 GOSUB 3120
5750 PRINT* INPUT Statements (cont)'
5760 PRINT
5770 PRINT 'Here is another example:'
5780 PRINT
5790 PRINT 'A = 20'
5800 PRINT '10 INPUT CHR$(24)"Enter a number between 1 and 9": IN'
5810 PRINT 'C = A: IN'
5820 PRINT 'RUN'
5830 PRINT
5840 PRINT 'Gives us: '
5850 PRINT 'Enter a number between 1 and 9: '
5860 PRINT 'If we enter a 5 then: '
5870 PRINT
5880 PRINT
5890 PRINT 'INPUT Statements (cont)'
5900 PRINT
5910 PRINT
5920 PRINT
5930 INPUT 'Press ENTER: TS'
5940 GOSUB 3120
5950 PRINT* INPUT Statements (cont)'
5960 PRINT
5970 PRINT 'INPUT Statements (cont)'
5980 PRINT 'We have learned then, that the INPUT statement allows you'
5990 PRINT 'to ENTER data in a program while it is running. It does'
6000 PRINT 'this by stopping the program and waiting for you to enter'
6010 PRINT 'data. When you do, it sets the data equal to the variable'
6020 PRINT 'on the end of the INPUT statement: '
6030 PRINT
6040 PRINT 'Between the message or prompt and the variable,'
6050 PRINT 'you must place a semi-colon (look at example above): '
6060 PRINT
6070 PRINT
6080 PRINT
6090 GOSUB 3120
Listing of Program 'LES02'

6470 PRINT* INPUT Statements*
6480 PRINT
6490 PRINT "You may use the INPUT statement without using a prompt or text."
6500 PRINT "If you do, then you must NOT put in a semi-colon."
6510 PRINT "For example:"
6520 PRINT
6530 PRINT 10 PRINT CHRS(34) "When you see a question mark, ENTER a 5" CHRS(34)
6540 PRINT 20 INPUT N*
6550 PRINT
6560 PRINT "When you see a question mark, ENTER a 5"
6570 PRINT?*
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 CHRS(34) "When you see a question mark, ENTER a 5" CHRS(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 PRINT CHRS(34) "ENTER a number between 1 and 9" CHRS(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."
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, put the right kind of variable on the end of the INPUT statement (you will be tested on this).

INPUT press ENTER:14
GOSUB 32
PRINT "INPUT Statements (cont)"
PRINT "PRINT Statements (cont)"
PRINT "RUN ENTER three (3) numbers"CHR$(34)"A,B,C"
PRINT "RUN ENTER three (3) numbers? 10"
PRINT "PRINT" "5"
PRINT "INPUT press ENTER:14"
GOSUB 32
PRINT "INPUT Statements (cont)"
PRINT "RUN ENTER three (3) numbers"CHR$(34)"A,B,C"
PRINT "RUN ENTER three (3) numbers? 10"
PRINT "PRINT" "5"
PRINT "INPUT press ENTER:14"
PRINT "PRINT Notice how the computer keeps prompting you with question marks. If you press ENTER without putting any data in, the computer will print out a character string of "A,B,C" until it gets all of its data. This can be a very useful feature.

PRINT ""
Listing of Program 'LESSON2' 07/10/83 - 00:01:19

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
7470 PRINT " ENTER three (3) numbers!"
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 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

160
Listing of Program 'LESSON2'

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$"
7920 PRINT
7930 PRINT"A 2,ten.15"
7940 PRINT"B 2,10,15 North Ela"
7950 PRINT"C TWO.1,22"
7960 PRINT"D 22,NONE,B"
7970 PRINT
7980 PRINT"ENTER the letter opposite the correct response and press ENTER":T$
7990 PRINT
8000 IF T$ = "B" GOTO 9300
8010 PRINT"WRONG - the correct answer is B (2,10,15 North Elma"
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) F. Lines 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" 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 STAND BY"
8372 RUN"LESSON2A"
8380 END
**** Listing of Program 'LESSON2A' ****

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 0 and then press ENTER": T$
340 IF T$ = "0" GOTO 7690
350 IF T$ = "C" GOTO 7700
360 IF T$ = "8" GOTO 430
370 IF T$ = "A" GOTO 280
380 GOSUB 550
390 GOSUB 5900
400 GOSUB 4720
410 GOSUB 6060
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 762
530 CN N GOSUB 550,3900,4720,6060,6640
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:*$'
850 GOSUB 6950
860 PRINT* READ and DATA (cont)*
870 PRINT
880 PRINT 'The format of the READ statement is:'
890 PRINT
900 PRINT '(line number) READ (variable or variables)*'
910 PRINT
920 PRINT 'An example of a READ statement that would read values into'
930 PRINT 'variables Bl, A, and X$ is:'
940 PRINT
950 PRINT '10 READ Bl,A,X$'
960 PRINT
970 PRINT 'INPUT 'press ENTER:*$'
980 GOSUB 6950
990 PRINT* READ and DATA (cont)*
1000 PRINT
1010 PRINT '10 READ Bl,A,X$'
1020 PRINT
1030 PRINT 'Note that both numeric and string variables may be "read"'
1040 PRINT 'An example of a DATA statement that would be read is:'
1050 PRINT
1060 PRINT '20 DATA 22,15."CHR$(34)"AIN'T she sweet?"CHR$(34)"
1070 PRINT
1080 PRINT 'Notice that the DATA statement has a different line number, but'
1090 PRINT 'it follows the same format as the READ statement. When these'
1100 PRINT 'two statements are placed in a program, the variables Bl, A, '
1110 PRINT 'and X$ would hold 22, 15, and "CHR$(34)"AIN'T she sweet?"CHR$(34)" respectively.*
1120 PRINT
1130 INPUT 'press ENTER:*$'
1140 GOSUB 6950
1150 PRINT* READ and DATA (cont)*
1160 PRINT
1170 PRINT* the two statements can appear anywhere in a program and in any
**Listing of Program 'LESSON2A'**

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":TS  
1270 GOSUB 650  
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":TS  
1350 PRINT  
1360 IF TS = "TRUE" THEN GOTO 1390  
1370 PRINT"WRONG - the sentence is valid."  
1380 GOTO 1400  
1390 PRINT"CORRECT"  
1400 PRINT  
1410 INPUT"press ENTER":TS  
1420 GOSUB 650  
1430 PRINT"READ and DATA (cont)"  
1440 PRINT  
1450 PRINT":0 READ B1,A.X$"  
1460 PRINT  
1470 PRINT"DATA 22.5,CHR$(24):AIN'T she sweet:"CHR$(24)  
1480 PRINT  
1490 PRINT"The variables in the READ statement, and the values in the"  
1500 PRINT"DATA statement are separated by commas, and the string is"  
1510 PRINT"enclosed in quotes. You cannot assign string data to a "  
1520 PRINT"numeric variable, if you do, your computer will throw up."  
1530 PRINT"Well, at the very least it will ALWAYS give you an ERROR"  
1540 PRINT"message."  
1550 PRINT  
1560 INPUT"press ENTER":TS  
1570 GOSUB 650  
1580 PRINT"READ and DATA (cont)"  
1590 PRINT":0 READ B1,A,Y"  
1600 PRINT"DATA 5.10,15"  
1610 PRINT"The average of these numbers is:"CHR$(24)"("  
1620 PRINT"(B1*AXY)/2"  
1630 PRINT"DATA 5.10,15"  
1640 PRINT"END"
1653 PRINT
1660 PRINT"Gives us:"
1670 PRINT
1680 PRINT5 10 15"
1690 PRINT"The average of these numbers is 10"
1700 PRINT
1710 PRINT"NOW! 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  1770 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 PRINT10 READ B1.A.Y* 1820 PRINT20 PRINT B1.A.Y*
1830 PRINT30 PRINT "CHR$(34)";"The average of these numbers is"CHR$(34)";"
1840 PRINT40 PRINT B1.A.Y*
1850 PRINT50 DATA 5,10,15"
1860 PRINT
1870 PRINT"Line 10 read the data in line 50. Line 20 PRINTed :t. 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$ 1925 GOSUB 6950
1930 PRINT  1940 PRINT" READ and DATA cont"
1940 PRINT
1950 PRINT 1960 PRINT"Let's look at it once more:"
1970 PRINT
2000 PRINT30 PRINT "CHR$(34)";"The average of these numbers is"CHR$(34)";"
2010 PRINT40 PRINT B1.A.Y*
2020 PRINT50 DATA 5,10,15"
2030 PRINT
2040 PRINT"Give us:"
2050 PRINT
2060 PRINT5 10 15"
2070 PRINT"The average of these numbers is 10"
2080 PRINT
2090 INPUT"press ENTER":T$ 2100 GOSUB 6950
**** Listing of Progarm 'LESSON2A' ****

2110 PRINT"Now for a neat example:"
2120 PRINT
2130 PRINT"IO READ A1.B1,C1"
2140 PRINT"2O PRINT A1.B1.C1"
2150 PRINT"SO 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 "ERROR" message that"
2230 PRINT"indicates we didn't have enough data for the number of vari-"
2240 PRINT"ables that we tried to READ."
2250 PRINT
2260 INPUT"press ENTER":T$
2270 GOSUB 550$
2280 PRINT"READ and DATA (cont)"
2290 PRINT
2300 PRINT"10 READ A1.B1,C1"
2310 PRINT"20 PRINT A1.B1.C1"
2320 PRINT"SO 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 550$
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$

107
***** Listing of Program 'LESSON2A' *****

2570 GOSUB 6750
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 2200
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 2830
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 out in it."
2820 GOTO 2840
2830 PRINT "CORRECT !"
2840 PRINT
2850 INPUT "press ENTER":T$
2860 GOSUB 6750
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 6750
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"
***** Listing of Program 'LESSONZA' *****

07/10/83 - 00:24:03

3020 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
3110 PRINT"10 PRINT "2"
3120 PRINT
3140 GOSUB 6450
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"
3210 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 6470
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 "TRUE"
3380 PRINT "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$
**Listing of Program 'LESSON2A'**

499 GOSUB 6950
500 PRINT "Read and DATA statements are used within a program, and they"
510 PRINT "do NOT stop the program so DATA can be entered."
520 PRINT
530 PRINT "A TRUE"
540 PRINT'S FALSE:
550 PRINT
560 INPUT 'press the letter opposite the correct answer and press ENTER':$'
570 PRINT
580 IF $ = 'A' THEN GOTO 6020
590 PRINT "W RONG - READ and DATA statements NOT called in the program, they"
600 PRINT "must be used within the program."
610 GOTO 8020
620 PRINT "CORRECT - you certainly have a good memory..."
630 PRINT
640 INPUT 'press ENTER: ':
650 GOSUB 6950
660 PRINT "READ A B C D"
670 PRINT 0 PRINT 0 PRINT 0 PRINT 0:
680 PRINT "RESTORE";
690 PRINT "READ C1, D1, E1"
700 PRINT 0 PRINT 0 PRINT 0:
710 PRINT 0 DATA 342, 24
720 PRINT
730 PRINT "The above program is ERROR free":
740 PRINT
750 PRINT "TRUE"
760 PRINT'S FALSE:
770 PRINT
780 INPUT 'ENTER the correct answer (either A or F)':
790 PRINT
800 IF $ = "A" GOTO 8340
810 PRINT "W RONG - Line 4, tried to read more DATA than was available."
820 PRINT "even though the RESTORE command was used."
830 GOTO 8320
840 PRINT "CORRECT:" 850 PRINT
860 INPUT 'press ENTER: ':
870 GOSUB 7100
880 IF $ = "G" THEN GOS 000
890 RETURN
900 GOSUB 0500
910 GOSUB 0500
920 PRINT "There are two types of branches, and we will be studying them" 930 PRINT "in the next two sections. They are CONDITIONAL branches, and"
940 PRINT "INCONDITIONAL branches. The above program has both kinds in it"
PRINT Line 30 is CONDITIONAL and line 40 is UNCONDITIONAL. Can you

PRINT*"see why? Study this for a moment and then press ENTER"!%

PRINT Line 30 is CONDITIONAL because it will only GOTO line 5; if
PRINT*"the CONDITION that A = 5 is satisfied. That is, control will"
PRINT*"only be transferred to line 5; if A = 5."
PRINT
PRINT Line 40 is UNCONDITIONAL because it will ALWAYS go to line 20"
PRINT*"when it is executed. There will be no choice made."
PRINT*"Control will go to line 20."
PRINT
PRINT
INPUT*"Press ENTER"!%
PRINT SUB 450
PRINT SUB 720
PRINT Notice that A will not equal 5 until line 20 is executed."
PRINT**"times. Therefore, until A = 5, the CONDITION in line 30 will*"
PRINT*"NOT be set and control will NOT GOTO line 50. Instead, it"
PRINT**"will go to the next line which is UNCONDITIONAL GOTO line 20."
PRINT
PRINT INPUT*"Study this carefully, and then press ENTER"!%
PRINT SUB 450
PRINT SUB 720
PRINT The statement in line 20 has made a COUNTER out of the variable" PRINT**"A. Every time the line is executed, A is incremented by one." PRINT**"COUNTERS are very useful in BASIC and we will discuss them" PRINT**"more in a future lesson. For now, try to understand how this" PRINT**"program works, and it will help you immensely in the future." PRINT
PRINT
PRINT INPUT*"Press ENTER"!%
PRINT SUB 450
PRINT PRINT The two types of branching are" PRINT**"if and GOTO."
PRINT**"IF A = 5 and GOTO*"
PRINT
PRINT INPUT*"Press the letter opposite the correct answer and press ENTER"!%
PRINT**"PRINT IF "A = 5 THEN GOTO 40:"* PRINT**"PRINT IF A = 5 THEN GOTO 40;" PRINT**"PRINT IF A = 5 THEN GOTO 40:* PRINT**"PRINT A = 5 THEN GOTO 40;"
***** Listing of Program "LESSON4" *****

**.410: GOTO 420**
**.420: PRINT"Correct"**
**.430: PRINT**
**.440: INPUT"Press ENTER",a**
**.450: GOSUB 420**
**.460: PRINT"H = ?"**
**.470: PRINT"IF A = 7 GOTO 5;"**
**.480: PRINT"H = A-1;"**
**.490: GOSUB 420; PRINT H**
**.500: PRINT"END"**
**.510: PRINT"END"**
**.520: PRINT**
**.530: PRINT"What would be the output of this program?"**
**.540: PRINT**
**.550: PRINT"A = ?"**
**.560: PRINT"B = ?"**
**.570: PRINT"No output would come from this program."**
**.580: PRINT"A = ?"**
**.590: PRINT**
**.600: INPUT"Press the letter opposite the correct answer and press ENTER",b**
**.610: PRINT**
**.620: IF b = 2 THEN GOTO 490**
**.630: PRINT"Wrong - the correct answer is 1.0;"**
**.640: GOSUB 450**
**.650: PRINT"Correct"**
**.660: PRINT**
**.670: INPUT"Press ENTER",a**
**.680: GOSUB 420**
**.690: IF b = 2 THEN GOTO 490**
**.700: RETURN**
**.710: GOSUB 430**
**.720: PRINT**
**.730: IF Statements"**
**.740: PRINT**
**.750: PRINT"IF Statements are decision makers in BASIC. The test to see"**
**.760: PRINT"IF a condition is met, IF it is, THEN the execute the command;"**
**.770: PRINT"IF statement does not follow the same line.**"**
**.780: PRINT**
**.790: PRINT"the IF statement causes the program to make decisions between"**
**.800: PRINT"true values. It is one of the most powerful commands in the BASIC"**
**.810: PRINT"language. You have already seen how it can be used in the"**
**.820: PRINT"previous section. In this section, we will explain it in a"**
**.830: PRINT"little more detail."**
**.840: PRINT**
**.850: IF b = 7; press ENTER;"**
**.860: END**
488) PRINT "This example shows three variations of the IF statement."
489) PRINT "IF we ENTER a '1' when prompted by line 10, then"
490) PRINT "the CONDITIONAL statement in line 20 will be satisfied."
491) PRINT "and the program will stop."
492) PRINT
493) INPUT "Please ENTER to continue":$"$121
494) GOSUB 121
495) PRINT
496) PRINT "IF we ENTER a '10' when prompted by line 10, the CONDITIONAL"
497) PRINT "statement in line 19 is not to be satisfied, nothing will happen."
498) PRINT until control passes to line 20. At that time, the check for"n
499) PRINT "IF number GREATER THAN 99 will be set, and the message will be printed."
500) PRINT "The other condition will be met until control sets to line 50."
501) PRINT "The unconditional GOTO at line 50 will send control back to the"
502) PRINT "beginning of the program."
503) PRINT
504) INPUT "Please ENTER to continue":$"$121
505) GOSUB 121
506) PRINT
507) PRINT "IF statement (cont)"n
508) PRINT "The IF statement can also be used to compare two expressions:"
509) PRINT "such as:"n
510) PRINT "PRINT IF 1+2=3 THEN GOTO 19"
511) PRINT "PRINT IT 2+2=4 THEN GOTO 19"
512) PRINT "PRINT "Assign variable assignment can be done in an IF statement:"
513) PRINT "PRINT IF 1=1 THEN A=1"
514) PRINT "PRINT IT 1=2 THEN A=2"
515) PRINT
516) PRINT
517) INPUT "Please ENTER to continue":$"$121
518) GOSUB 121
519) PRINT "There is another word that can be added to the IF statement:"
520) PRINT "ELSE to make it more powerful. It is the ELSE word:"n
521) PRINT "PRINT IF A=1 THEN GOTO 20 ELSE GOTO 30"
522) PRINT "PRINT IT A=2 THEN GOTO 30 ELSE GOTO 40"
523) PRINT
524) PRINT "In this line, if the variable A equals 1 then control transfers:
**** Listing of Program 'LETRONIC' ****

5240 PRINT "to line 10: if it does NOT equal 1 then control transfers to 1"  
5250 PRINT "to line 100. In this case, something ALWAYS happens at line 120."
5260 PRINT "because of the ELSE statement."
5270 PRINT
5280 PRINT "GOTO 100 IF A = 1 THEN GOTO 10 ELSE IF A = 2 GOTO 20"  
5290 PRINT
5300 PRINT "In this case, if A = 1 OR A = 2 then something will happen."
5310 PRINT "In line 120, if none of the CONDITIONS are met, then the line:"  
5320 PRINT "will not be executed."
5330 PRINT
5340 PRINT
5350 INPUT 'Press ENTER to: #'  
5360 INPUT "A" $150  
5370 PRINT: IF Statement (cont)"
5380 PRINT
5390 PRINT: "The IF statement instructs the computer to test the following"  
5400 PRINT: "logical or relational expression. If the expression is TRUE"  
5410 PRINT: "then control will proceed to the action line after the THEN"  
5420 PRINT: "word. If the expression is not true, then control will proceed"  
5430 PRINT: "to the ELSE action."
5440 PRINT
5450 INPUT 'Press ENTER to: #'  
5460 INPUT 'A' $150  
5470 PRINT: "What is the output of the following program?"  
5480 PRINT
5490 PRINT: "100 A = 255"  
5500 PRINT: "200 IF A = 100 THEN PRINT "CHR$(49)" "TOO WEAK" "CHR$(49)"
5510 PRINT: "200 IF A = 255 THEN PRINT "CHR$(54)" "TOO STRONG" "CHR$(54)"
5520 PRINT: "400 IF A = 255 THEN PRINT "CHR$(49)" A = 255 "CHR$(49)"
5530 PRINT
5540 PRINT: "A = 255 "  
5550 PRINT: "A = 255"  
5560 PRINT: "A = 255"  
5570 PRINT: "END"  
5580 PRINT: "A = 255"  
5590 PRINT: "A = 255"  
5600 PRINT
5610 PRINT: "INPUT 'Enter the correct answer':"  
5620 INPUT 'A' $150  
5630 IF A = "C" GOTO 5700
5640 PRINT: "The correct answer is C (TOO STRONG):"  
5650 PRINT: "A = 255:"  
5660 GOTO 5700
5670 PRINT: "The correct - we need our mind in the budget office!"
**** Listing of Program "LESSON9" ****

5790 PRINT
5800 INPUT": press ENTER":"1%
5810 GOSUB 5750
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":1%
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":1%
5970 GOSUB 7100
5980 IF T% = "B" GOTO 4720
5990 RETURN
6000 GOSUB 5950
6010 PRINT"GOTO Statements"
6020 PRINT
6030 PRINT Conditional branches are written with IF THEN FALSE
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:"1%
6080 GOSUB 4710
6090 PRINT
6100 INPUT": press ENTER":1%
6110 GOSUB 4720
6120 GOSUB "A"
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":1%
6180 GOSUB "B"
6190 PRINT"GOTO (cont.)"
6200 PRINT
6210 PRINT."You can make the GOTO statement a MULTI-way branching statement"
6220 PRINT"by adding it slightly. For example:"1%
6230 PRINT
6240 GOSUB "C"
***** Listing of Program 'LESSON2B' *****

250 PRINT
260 INPUT"press ENTER":":
270 GOSUB 6950
280 GOSUB 7580
290 PRINT
300 PRINT"When line 20 is executed, the value of N is used to count over"
310 PRINT"N" elements passed the GOTO word. Control branches to the"
320 PRINT"line number indicated by this "Nth" element. If there is no"
330 PRINT"element that corresponds to the value of N, then control passes"
340 INPUT"to the next available line. Press ENTER when ready":
350 GOSUB 6950
360 GOSUB 7580
370 PRINT
380 PRINT"The value of N MUST be greater than 0 and less than 255. If it"
390 PRINT"is not, BASIC will print an error. If N is 1, 2, or 3 then the"
400 PRINT"program will print the appropriate message and stop."
410 PRINT
420 INPUT"press ENTER":":
430 GOSUB 6950
440 PRINT"Which lines (beyond 20) are executed if you ENTER a 10 here":
450 PRINT
460 GOSUB 7580
470 PRINT
480 PRINT"A ""50 and 40"
490 PRINT"B ""50 and 60"
500 PRINT"C ""70 and 30"
510 PRINT
520 INPUT"press the letter opposite the correct answer then press ENTER":":
530 PRINT
540 IF $ = "A" GOTO 570
550 PRINT"WARNING - the correct answer is A (50 and 40)"
560 GOTO 6570
570 PRINT"CORRECT"
580 PRINT
590 INPUT"press ENTER":":
600 GOSUB 6950
610 PRINT"You are now done with this lesson. When you hit ENTER, your"
620 PRINT"will be returned to the MENU where you may review sections"
630 PRINT"or take the TEST."
640 PRINT
650 PRINT"Remember, after you are done here, practice some of the"
660 PRINT"things you have learned. And keep a BASIC manual by your side."
670 PRINT"This program will show you the fundamentals, you have to teach"
680 PRINT"yourself how to be good at BASIC. That means you must PRACTICE"
690 PRINT

110
### Listing of Program 'LESSON 2A'

**INPUT** press ENTER to go to the MENU: **

RUN

REM ** This subroutine clears the screen on any terminal
REM **

FOR I = 1 TO 24
PRINT
NEXT I
RETURN

PRINT  
PRINT ** This is the second part of a two part lesson **
PRINT ** It is divided into the following sections: **
PRINT **
PRINT 1) READ, DATA and RESTORE **
PRINT 2) Branching Introduction **
PRINT 3) IF Statements **
PRINT 4) GOTO Statements **
PRINT 5) TEST **
PRINT
PRINT
RETURN

GOSUB 6950
PRINT ** Which do you want to do? **
PRINT
PRINT ** Continue on **
PRINT
PRINT ** Review this lesson again **
PRINT
PRINT ** press the letter opposite your choice and press ENTER **: **
PRINT 1) IF **
PRINT 2) **
PRINT 3) END **
PRINT
RETURN

REM ** This subroutine prints the Branching section example **
REM
PRINT ** Branching Introduction **
PRINT
PRINT 10 A = 0
PRINT 20 A = A + 1
PRINT 30 IF A = 5 THEN GOTO 50
PRINT 40 GOTO 20
PRINT 50 PRINT A
PRINT 60 END
PRINT
PRINT
RETURN
REM
REM ** This subroutine is for the IF statement examples **
REM
PRINT ** IF Statements (cont) **
**** Listing of Program 'LESSON2' ****

07/10/83 - 00:24:03

7360 PRINT
7370 PRINT"10 INPUT "CHR$(4)"ENTER a number between 1 and 10 to quit:"CHR$(4)"N"
7380 PRINT"20 IF N = 0 THEN STOP"
7390 PRINT"30 IF N > 10 THEN PRINT "CHR$(4)"ERROR - you entered an invalid number"CHR$(4)"
7400 PRINT"40 IF N = 0 PRINT "CHR$(4)"ERROR - you entered an invalid number"CHR$(4)"
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$(4)"ENTER a number between 1 and 10"CHR$(4)"N"
7490 PRINT"20 IF N = 8 THEN GOTO 50"
7500 PRINT"30 IF N = 9 THEN PRINT "CHR$(4)"GUESS AGAIN"CHR$(4)"
7510 PRINT"40 GOTO 10"
7520 PRINT"50 PRINT "CHR$(4)"YOU GUessed IT"
7530 PRINT"60 END"
7540 RETURN
7550 REM
7560 REM this is the subroutine example for ON GOTO
7570 REM
7580 PRINT"10 INPUT "CHR$(4)"ENTER a number between 1 and 3"CHR$(4)"N"
7590 PRINT"20 ON N GOTO 20,50,70"
7600 PRINT"30 PRINT "CHR$(4)"YOU ENTERED A ONE (or a illegal number)"CHR$(4)"
7610 PRINT"40 STOP"
7620 PRINT"50 PRINT "CHR$(4)"YOU ENTERED A TWO"CHR$(4)"
7630 PRINT"60 STOP"
7640 PRINT"70 PRINT "CHR$(4)"YOU ENTERED A THREE"CHR$(4)"
7650 PRINT"80 STOP"
7660 RETURN
7670 REM
7680 RUN"TEST2"
7690 RUN"MENU"
7700 RUN"LESSON2"
7710 END
**** Listing of Program 'TEST2' ****

10 REM **
20 REM ** LESSON: TEST2 VERSION: 1 AUG 83
30 REM ** AUTHOR: CAPT DAN CREGAN
40 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
50 REM **
60 REM ** VARIABLES:
70 REM ** N$(1) = NAMES ARRAY, USED TO READ IN SEQUENTIAL NAMES, AND TO WRITE OUT
80 REM ** UPDATE NAMES.
90 REM ** S$(1) = SCORES ARRAY - USED TO READ AND WRITE SCORES
100 REM ** Q$(1) = ARRAY TO KEEP TRACK OF NUMBER OF CORRECT ANSWERS, IF AN ARRAY ELEMENT EQUALS 1, THE ANSWER WAS CORRECT.
110 REM ** TO UPDATE NAMES.
120 REM ** VARIABLES:
130 REM ** SCORE$(1) - ARRAY TO KEEP TRACK OF NUMBER OF CORRECT ANSWERS.
330 PRINT* 'FINAL TEST: LESSON 2'
340 PRINT 'This test consists of 10 questions. You must get 70 percent to pass. (That's 7 right out of 10 questions). Use only capital letters in your answers, don't include extra spaces or letters. GOOD LUCK'
350 PRINT 'Press ENTER to continue: '
360 GOSUB 2281
370 PRINT 'Which of the following is a legal filename?'
380 PRINT A 'SODIIIE.COM' (in CPM or Cromemo)
390 PRINT B 'TRIUMPH.SCI' (in TRS-80)
400 PRINT C 'THEWAYOF.S21' (in CPM or Cromemo)
410 PRINT D 'SCORE7.OAT' (in TRS-99)
420 PRINT 'Enter the letter opposite the correct answer: '
430 IF T$ = 'C' THEN GOTO 470
440 PRINT 'Wrong - the correct answer is C'
450 PRINT 'Answers A and B filenames or extensions start with numbers instead of alphabet characters. Answer D has a non-alphanumeric character in it.'
460 PRINT 'See part 1, filenames, in Lesson 2.'
470 GOTO 490

119
LISTING OF PROGRAM 'TEST2'

470 PRINT 'CORRECT'
480 G1I:= 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 G1I:= 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 G1I:= 1
880 PRINT
890 INPUT 'press ENTER':T$
900 GOSUB 3280
910 PRINT 'Which of the following examples is INVALID'
920 PRINT
**Listing of Program 'TEST2'**

07/10/83 - 00:48:52

930 PRINT"A" INPUT T$*
940 PRINT"B" INPUT CHRS(34)"ENTER YOUR NAME"CHR$(34)"IN"
950 PRINT"C" INPUT N*
960 PRINT"D" INPUT CHRS(34)"ENTER YOUR AGE"CHR$(34)"IN"
970 PRINT
980 INPUT"ENTER the letter opposite the correct answer":T$*
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 I."
1040 GOTO 1070
1050 PRINT"CORRECT"
1060 Q1A = 1
1070 PRINT
1080 INPUT"press ENTER":T$*
1090 GSUB 3280
1100 PRINT"What command will let you put your program into permanent"
1110 PRINT"storage so that you can recall it later?"
1120 PRINT
1130 PRINT"A" SAVE*
1140 PRINT"B" RUN*
1150 PRINT"C" LOAD*
1160 PRINT"D" STORE*
1170 PRINT
1180 INPUT"ENTER the letter opposite the correct answer":T$*
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 I."
1250 GOTO 1280
1260 PRINT"CORRECT"
1270 Q1B = 1
1280 PRINT
1290 INPUT"press ENTER":T$*
1300 GSUB 3280
1310 PRINT"Which of the following statements is legal?"
1320 PRINT
1330 PRINT"A" READ A B C*
1340 PRINT"B" DATA A B C*
1350 PRINT"C" READ C, C, C*
1360 PRINT"D" DATA "CHR$(34)"QUIT"CHR$(34)"."CHR$(34)"*FIRE"CHR$(34)"
1370 PRINT
1380 INPUT"ENTER the letter opposite the correct answer":T$*
Listing of Program 'TEST2'

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(9) = 1
1840 PRINT
**** Listing of Program 'TEST2' ****
07/10/83 - 09:48:52

1950 INPUT "press ENTER":?$
1960 GOSUB 2260
1970 PRINT "The following program will NOT have an output - TRUE or FALSE"
1980 PRINT
1990 PRINT "IC N = 5"
2000 PRINT "GO TO 40,50,80"
2010 PRINT "STOP"
2020 PRINT "END"
2030 IF $ = "A" GOTO 2090
2040 PRINT "WROONG - 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 $ = "$"""
2110 PRINT
2120 INPUT "press ENTER":?$
2130 GOSUB 2260
2140 PRINT "The following program will have an output - TRUE or FALSE"
2150 PRINT
2160 PRINT "IC N = 3"
2170 PRINT "GO TO 40,50,80"
2180 PRINT "IF N = 3 THEN GOTO 40"
2190 PRINT "END"
2200 PRINT "END"
2210 PRINT
2220 PRINT "A TRUE"
2230 PRINT "$ FALSE"
2240 PRINT
2250 PRINT
2260 INPUT "Enter the letter opposite the correct answer":?$
2270 PRINT
2280 IF $ = "$" GOTO 2340
2290 PRINT "WRONG - the correct answer is $"
2300 PRINT Line 10 sets N to 3. line 20 causes the program to"
**** Listing of Program 'TEST2' ****

2210 PRINT* go to line 40, then 50.*
2220 PRINT* See part 1, IF.*
2230 GOTO 2250
2240 PRINT"CORRECT"?
2250 G10+ 1
2260 PRINT
2270 INPUT"press ENTER":T$
2280 GOSUB 3280
2290 FOR I = 1 TO 10
2300 Y = Y + G1(X)
2310 NEXT I
2320 PRINT"You have finished the test, out of 10 possible correct answers"*
2330 PRINT"You scored "Y."*
2340 PRINT
2350 IF Y > 6 THEN PRINT"YOU HAVE PASSED"
2360 GOSUB 3110
2370 IF Y > 6 THEN GOTO 2550
2380 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"*
2390 PRINT
2400 PRINT"YOU SHOULD RETAKE LESSON 2!"
2410 PRINT
2420 PRINT"You will be returned to the Menu."*
2430 PRINT
2440 GOTO 3300
2450 PRINT
2460 PRINT"Do you want your score recorded on a permanent file?"
2470 PRINT
2480 PRINT'A' YES*'
2490 PRINT'B' NO
2500 PRINT
2510 INPUT"Which":T$
2520 IF T$ = "A" THEN GOTO 2920
2530 GOSUB 3220
2540 PRINT"To record your score, we must open a file and put your name"*
2550 PRINT"in it. Therefore, surprisingly, we need your name. If your"*
2560 PRINT"name is not unique among the students likely to take this test,"*
2570 PRINT"please contact your test monitor for an identifying word that"*
2580 PRINT"will make you unique. Then enter that word below."
2590 PRINT
2600 PRINT"If you have already entered a score previously, be sure to"*
2610 PRINT"enter the same name you used before. Just all capitals!"
2620 PRINT
2630 INPUT"ENTER your word or name now":T$
2640 OPEN*",1,"SCORE2"*
2650 I = 0
2660 IF EOF(I) THEN GOTO 2920
2770 \( i = i+1 \)
2780 INPUT #1, \( N(i) \);
2790 PRINT #1, \( S(i) \);
2800 IF \( N(i) = 1 \) THEN GOTO 2970
2900 GOTO 2750
2910 \( i = i+1 \)
2920 \( N(i) = 1 \)
2930 \( S(i) = Y \)
2940 OPEN #1, "SCORE2"
2950 FOR \( W = 1 \) TO \( I \)
2960 PRINT #1, \( N(W) \)
2970 PRINT #1, \( S(W) \)
2980 NEXT \( W \)
2990 PRINT
3000 PRINT "You are now qualified to go to LESSON 3, however, you will"
3010 PRINT "first get a homework assignment! The homework is in a "
3020 PRINT "program module. If you don't want the assignment, you may"
3030 PRINT "go to the MENU instead."
3040 GOTO 3300
3050 \( S(i) = Y \)
3060 IF EOF(1) THEN CLOSE; GOTO 3200
3070 \( i = i+1 \)
3080 INPUT #1, \( N(i), S(i) \)
3090 GOTO 2970
3100 OPEN #1, "SCORE2"
3110 FOR \( W = 1 \) TO \( I \)
3120 PRINT #1, \( N(W) \)
3130 PRINT #1, \( S(W) \)
3140 NEXT \( W \)
3150 PRINT
3160 PRINT "You may now take LESSON 3. You will be returned to the MENU"
3170 PRINT "from where you may go to LESSON 3 or quit."
3180 GOTO 3300
3190 IF \( Y = 1 \) THEN RETURN
3200 PRINT "YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
3210 PRINT
3220 IF \( Q(1) = 0 \) THEN PRINT "part 1. Filenames"
3230 IF \( Q(2) = 0 \) OR \( Q(5) = 0 \) THEN PRINT "part 1. SAVE, LOAD, RUN"
3240 IF \( Q(6) = 0 \) THEN PRINT "part 1. Remarks"
3250 IF \( Q(7) = 0 \) THEN PRINT "part 1. INPUT Statements"
3260 IF \( Q(8) = 0 \) OR \( Q(7) = 0 \) THEN PRINT "part 2. READ and DATA Statements"
3270 IF \( Q(9) = 0 \) OR \( Q(8) = 0 \) THEN PRINT "part 2. IF Statements"
3280 IF \( Q(10) = 0 \) THEN PRINT "part 2. GOTO Statements"
3290 PRINT
3300 INPUT "Press ENTER: ":$
***** Listing of Program 'TEST2' *****

1000 GOSUB 1080
1140 REM RETURN
1150 REM **
1160 REM ** this subroutine clears the screen**
1170 REM **
1180 FOR X = 1 TO 24
1190 PRINT
1200 NEXT X
1210 RETURN
1220 PRINT
1230 INPUT "Press ENTER to return to the MENU":$1$2
1240 RUN "MENU"
1250 PRINT
1260 PRINT
1270 INPUT "Do you want the assignment (Y/N)":$2$3
1280 IF $2$3 = "N" THEN GOTO 1240
1290 RUN "HW2"
***** Listing of Program 'LESSON3' *****

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 J  VERSION: 1 AUGUST 93"
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": T1
250 GOSUB 5590
260 GOSUB 5660
270 PRINT "I'm taking this part in its entirety."
280 PRINT "I wish to review selected areas."
290 PRINT "I want to go to the second part."
300 PRINT "I want to return to the Menu."
310 PRINT
320 INPUT "Press either capital A, B, C, or D and then press ENTER": T1
330 IF T1 = "A" GOTO 6370
340 IF T1 = "B" GOTO 3360
350 IF T1 = "C" GOTO 460
360 IF T1 = "D" GOTO 370
370 GOSUB 370
380 GOSUB 1090
390 GOSUB 1450
400 GOSUB 1240
410 GOSUB 3430
420 PRINT
430 PRINT
440 PRINT "Going to second half of lesson 3 - Wait one moment."
450 GOTO 3380
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 570
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 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."
730 PRINT
740 INPUT "press ENTER to learn about the KILL command": N
750 GOSUB 5590
760 PRINT "Introduction & KILL (cont)"
770 PRINT
780 PRINT "The format for the KILL command is:"
790 PRINT "KILL "CHR$(34)"filename.ext"CHR$(34)"
800 PRINT
810 PRINT "You must enclose the filename in quotation marks."
820 PRINT "The extension is only needed when the original file had one."
830 PRINT
840 PRINT "Once this command is entered, the file will be removed unless"
850 PRINT "protected by passwords - you can learn about passwords from"
860 PRINT "your system manual). If the file is removed, there is very"
870 PRINT "little chance of recovering any of your data. In some cases"
880 PRINT "an advanced programmer can retrieve data from a killed file"
890 PRINT "but the process is difficult and often fails. BE CAREFUL!"
900 PRINT
910 INPUT "press ENTER": N
***** Listing of Program "LESSON3" *****

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":TS
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":TS
1060 GOSUB 5290
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":TS
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 9, 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 I = 1 TO 9
1290 PRINT INT(I), INT(I^2), INT(I^2 + I)
1300 NEXT I
1310 PRINT
1320 INPUT "press ENTER":TS
1330 GOSUB 5590
1340 FOR I = 1 TO 9
1350 PRINT INT(I), INT(I^2), INT(I^2 + I)
1360 NEXT I

129
Listing of Program 'LESSON3' 07/10/91 - 01:02:38

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":?%
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 6590
1540 PRINT "The value of I after running this program is 10."
1550 PRINT
1560 INPUT "press ENTER":?$
1570 GOSUB 5590
1580 PRINT "COUNTERs (cont)"
1590 PRINT
1600 GOSUB 6590
1610 PRINT "The COUNTER variable is I. It is INCREMENTED every time line"
1620 PRINT "40 sends control to it. When the CONDITIONAL statement is:"
1630 PRINT "line 30 discovers that I is equal to 10, it executes the STOP."
1640 PRINT "At the end of the program, I is equal to 10."
1650 PRINT
1660 INPUT "press ENTER":?$
1670 GOSUB 5590
1680 GOSUB 6590
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 "I=0"
1730 PRINT "I=I+1"
1740 PRINT "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, HANG, and *** DUMMY, you did it again! ***"
1790 PRINT
1800 INPUT "press ENTER":?$
1810 GOSUB 5590
1820 PRINT "COUNTERs cont:"

129
***** Listing of Program 'LESSON3' *****

**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** "Do 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 1) **"$
**1880 PRINT** "CAUTION, if you are using this program on a multi-user system. **"$
**1890 PRINT** "such as Cromemco System II, DO NOT REST 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**

**1970 DDBUS SENT**

**1980 PRINT** "What is the COUNTER variable in this program?**

**1990 PRINT**

**2000 PRINT** "10 LET**

**2010 PRINT** "20 X**

**2020 PRINT** "30 X**

**2030 PRINT** "IF I = 0 THEN I = 2**

**2040 PRINT** "IF I = 5 THEN STOP**

**2050 PRINT** "PRINT I**

**2060 PRINT** "GOTO 20**

**2070 PRINT** "END**

**2080 PRINT**

**2090 PRINT** "I is the COUNTER**

**2100 PRINT** "X is the COUNTER**

**2110 PRINT** "BOTH are COUNTERS, but X will control the program**

**2120 PRINT**

**2130 INPUT*** press the letter opposite the correct answer and press ENTER**

**2140 PRINT**

**2150 IF I = 9 THEN GOTO 218**

**2160 PRINT** "INCORRECT - the correct answer is B**

**2170 GOTO 110**

**2180 PRINT** "CORRECT - that shows good understanding of the principle**

**2190 PRINT**

**2200 INPUT*** press ENTER**

**2210 IF I = 9 THEN GOTO 145**

**2220 PRINT** "FOR - NEXT Statements**

**2230 PRINT**

**2240 PRINT** "END of the previous discussion of counters. We showed you that we**
**** Listing of Program 'LESSON' ****

1220 PRINT "tests a counter with an IF statement to see if it has reached"
1230 PRINT "the desired level, if it had then we went to another part of the"
1240 PRINT "program, or STOPed. There is a set of statements in BASIC"
1250 PRINT "that lets us shorten the statements needed to duplicate this:
1260 PRINT "AND of program. The set of statements is called FOR-NEXT."
1270 PRINT
1280 PRINT
1290 INPUT "press ENTER":
1295 GOSUB 5790
1300 PRINT "FOR-NEXT (cont)"
1305 GOSUB 5790
1310 PRINT "The two programs above will print exactly the same output."
1315 PRINT "The COUNTER loop requires more statements and is not as"
1320 PRINT "efficient as the FOR NEXT loop."
1325 PRINT
1330 INPUT "press ENTER":
1335 GOSUB 5790
1340 GOSUB 5790
1350 PRINT "Line 10 of the FOR NEXT loop initializes i, the same way that"
1355 PRINT "lines 10 and 20 of the COUNTER loop do. The FOR NEXT loop sees"
1360 PRINT "line 10 as "I'm going to start a loop. i is the counter. I will"
1365 PRINT "begin with i = 1 and when it is GREATER than 3 I will GOTO the"
1370 PRINT "statement that is after the NEXT i statement.""
1375 PRINT
1380 INPUT "press ENTER":
1385 GOSUB 5790
1390 GOSUB 5790
1395 GOSUB 5790
1405 PRINT "Line 20 of the COUNTER loop and Line 30 of the FOR NEXT loop."
1410 PRINT "are the same and perform the same function."
1415 PRINT "Note that the PRINT statement was indented a few spaces in the"
1420 PRINT "FOR NEXT loop. This lets you see the loop structure better."
1425 PRINT "You should do that in your own programming, too."
1430 PRINT
1435 INPUT "press ENTER":
1440 GOSUB 5790
1445 GOSUB 5790
1450 PRINT "The IF statement in line 40 of the COUNTER loop determines if i"
1455 PRINT "has reached 3. This was done automatically by the FOR NEXT"
1460 PRINT "loop because the last value in line 40 was specified as 3.""
1465 PRINT "Line 50 of the FOR NEXT loop is the same as the UNCONDITIONAL"
1470 PRINT "GOTO in line 50 of the COUNTER loop. The NEXT i statement"
1475 PRINT "increments i and sends control to the FOR statement."
1480 PRINT "In this case control goes to line 1. press ENTER""
1485 INPUT "press ENTER":
1490 GOSUB 5790
1495 GOSUB 5790
1500 PRINT "IF statement"
"LESSON:

17. PRINT: "DO NEXT";
18. PRINT;
19. PRINT: "In assembler, the FOR X = value #1 to value #2";
20. PRINT: "causes the variable X to be initially set at value #1.";
21. PRINT: "and the program executes the next lines until it reaches the"
22. PRINT: "NEXT" statement. It is an example of a variable worked. The"
23. PRINT: "NEXT" statement causes an unconditional GOTO to the FOR state-
24. PRINT: "ment. The variable is incremented and the FOR word tests to"
25. PRINT: "see if it exceeds value #2. IF it does, then control passes"
26. PRINT: "to the statement that follows the NEXT statement.";
27. PRINT: "there are no statements passed the NEXT word, the program"
28. PRINT: "continues.";
29. PRINT;
30. INPUT: "Press ENTER";
31. GOSUB 5590;
32. PRINT: "FOR NEXT (cont)";
33. PRINT;
34. PRINT: "This concludes the initial FOR NEXT section. You should under-
35. PRINT: "stand what a simple FOR NEXT statement does. If you do not,"
36. PRINT: "you will be given a chance to review this section before you"
37. PRINT: "proceed on. But before that - still timer.";
38. PRINT;
39. INPUT: "Press ENTER";
40. GOSUB 5590;
41. FOR I = 1 TO 10;
42. PRINT: "X = I";
43. NEXT I;
44. PRINT;
45. PRINT: "Which of the following programs make the above list?"
46. PRINT;
47. PRINT: 1. FOR I = 1 TO 10 FOR I = 1 TO 10
48. PRINT: 20 PRINT X = I 20 PRINT X = I
49. PRINT: 30 NEXT I 30 NEXT I
50. PRINT;
51. PRINT: 1. IF X = 1 THEN GOTO 1 1. IF X = 1 THEN GOTO 1
52. PRINT: 20 GOTO 40 20 GOTO 10
53. PRINT;
54. INPUT: "Press the letter opposite the correct choice and press ENTER";
55. PRINT;
56. IF "A" = "A" THEN GOTO 10
57. PRINT: "Wrong - the correct answer is A";
58. GOTO 10;
59. PRINT: "Correct";
60. PRINT;
INPUT press ENTER
Goes to 229
PRINT "The NEXT statement is the same as an UNCONDITIONAL BRANCHING"
PRINT "Statement. It branches to the next line under the FOR statement."
PRINT "enter.
PRINT "Is the paragraph above TRUE or FALSE?"
PRINT "TRUE"
PRINT "FALSE"
PRINT "INPUT the letter opposite the correct answer: T or F"
IF "T" = "T" THEN GOTO 229
PRINT "F""RONG - the NEXT statement is UNCONDITIONAL. It does not branch"
PRINT "STATEMENT to the same line as the FOR statement. It also increments the variable before the FOR statement tests it."
GOTO 322
PRINT "CORRECT"
PRINT "INPUT press ENTER"
GOSUB 322
IF "F" = "F" THEN GOTO 229
RETURN
GOSUB 356
PRINT "Advanced FOR NEXT"
PRINT "FOR i = 1 TO 10 STEP 2"
PRINT "NEXT i"
PRINT "Goes up"
PRINT "FOR i = 1 TO 10 STEP 2"
PRINT "NEXT i"
PRINT "FOR i = 1 TO 10 STEP 2"
PRINT "NEXT i"
PRINT "The STEP word after the FOR statement caused the value of i to
Listing of Program 'LESON3'  

1610 PRINT "The output of 1,2,3,4 was"  
1620 PRINT "correct because the NEXT statement is the statement that"  
1630 PRINT "actually increments the value, so the first time through,"  
1640 PRINT "was equal to 1, the next time it was equal to 2, etc."  
1650 PRINT  
1660 INPUT "press ENTER for explanation"  
1670 GOSUB 3590  
1680 PRINT "FOR x = 10 to 1 STEP -2"  
1690 PRINT "NEXT x"  
1700 PRINT  
1710 PRINT "Because the value of the FOR statement is the largest value, it"  
1720 PRINT "would not make sense to specify a negative STEP and give the"  
1730 PRINT "range of a positive STEP. For example:"  
1740 PRINT  
1750 PRINT "PRINT in this case, we step DOWN instead of up. Notice that the"  
1760 PRINT "first value of the FOR statement is the largest value. It"  
1770 PRINT "the first value of the FOR statement is the largest value. It"  
1780 PRINT "would not make sense to specify a negative STEP and give the"  
1790 PRINT "range of a positive STEP. For example:"  
1800 PRINT  
1810 PRINT "FOR x = 1 to 10 STEP -1"  
1820 PRINT  
1830 PRINT "This doesn't make sense, and would not work."  
1840 PRINT  
1850 INPUT "press ENTER for explanation"  
1860 GOSUB 3590  
1870 PRINT "You may include a FOR NEXT statement within another FOR NEXT"  
1880 PRINT "statement. If you do, it is called NESTED looping."  
1890 PRINT "For example:"  
1900 GOSUB 3590  
1910 PRINT "Before we explain the output, do you see how we indented the"  
1920 PRINT "statements within the first loop, and then further indented"  
1930 PRINT "the statements within the second loop? It makes the easier"  
1940 PRINT "to understand (and explain)."  
1950 PRINT  
1960 INPUT "press ENTER for output"  

Note: The output of 1,2,3,4 was correct because the NEXT statement is the statement that actually increments the value, so the first time through, it was equal to 1, the next time it was equal to 2, etc. Because the 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: 

FOR x = 10 to 1 STEP -2
NEXT x

PRINT in this case, we step 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: 

FOR x = 1 to 10 STEP -1

This doesn't make sense, and would not work. 

You may include a FOR NEXT statement within another FOR NEXT statement. If you do, it is called NESTED looping. For example: 

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 the easier to understand (and explain).
1100 GOSUB 5590
1120 PRINT"Advanced FOR NEXT (NESTED LOOPS):"
1110 PRINT
1140 FOR X = 1 TO 2
1150 FOR Y = 1 TO 2
1160 PRINT X,Y
1170 NEXT Y
1180 NEXT X
1190 PRINT
1200 GOSUB 5590
1210 PRINT"Try to figure now how the program produced the columns above it"
1220 INPUT"and then press ENTER":T$
1230 GOSUB 5590
1240 GOSUB 5290
1250 PRINT"On the first pass through the program, the FOR NEXT loop for"
1260 PRINT"X was set up, and control passed to the lines between FOR X ="
1270 PRINT"to 2, and NEXT X. Those lines happened to be another loop"
1280 PRINT"with the variable Y as the FOR NEXT variable. When the Y FOR"  
1290 PRINT NEXT executed the first time, X was equal to 1 and Y was"
1300 PRINT"printed twice, once as a 1 and once as a 2. When the Y loop"
1310 PRINT"finished, the NEXT X statement caused control to GOTO line"
1320 PRINT"16 again. The process was repeated, but X now was equal to 2."
1330 PRINT
1340 GOSUB 5590
1350 PRINT"as for Y
1360 FOR X = 1 TO 2
1370 FOR Y = 1 TO 2
1380 PRINT X,Y
1390 NEXT Y
1400 NEXT X
1410 PRINT
1420 PRINT FOR X = 1 TO 2
1430 PRINT FOR Y = 1 TO 2
1440 PRINT PRINT X,Y
1450 PRINT NEXT Y
1460 PRINT FOR NEXT X"
1470 PRINT
1480 INPUT"Can you figure it out now? Think about it and then press ENTER":T$
1490 GOSUB 5590
1500 GOSUB 5290
1510 PRINT"The most common mistake that programmers make when using nested"
1520 PRINT"loops is mislabeling the NEXT statements. Notice that the"
1530 PRINT"NEXT statement for the i variable is placed before the NEXT X"
1540 PRINT"statement."
1550 PRINT
1560 INPUT
1570 GOSUB press ENTER":T$

136
***** Listing of Program 'LESSON3' *****

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:IT"
4700 GOSUB 5590
4710 PRINT"Is the following program valid?"
4720 PRINT
4730 PRINT"FOR I = 1 TO 20"
4740 PRINT" FOR J = 1 TO 20"
4750 PRINT" I = J"
4760 PRINT" NEXT J"
4770 PRINT" NEXT I"
4780 PRINT
4790 PRINT"A YES it is a good program"
4800 PRINT"NO it is not a valid program"
4810 PRINT
4820 INPUT"ENTER the letter opposite the correct answer:IT"
4830 PRINT
4840 IF * = "A" GOTO 4850
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:IT"
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:IT"
4980 PRINT
4990 IF * = "A" FOR I = 1 TO 20 STEP 2 GOTO 5330
5000 PRINT"WRONG - the answer is:"'
5010 PRINT" 10 FOR I = 1 TO 20 STEP 2"
5020 GOTO 594.
***** Listing of Program 'LESSON3' *****

5010 PRINT"CORRECT! - You are a winner!"
5040 PRINT
5150 INPUT"press ENTER":TS
5060 GOSUB 5590
5970 PRINT"is the following program valid?"
5090 PRINT
5390 PRINT"FOR I = 1 TO 2"
5100 PRINT":FOR I = 1 TO 5"
5110 PRINT"PRINT CHR$(34):""This is the last question"CHR$(34)
5120 PRINT"NEXT I"
5130 PRINT"NEXI X"
5140 PRINT
5150 PRINT"Yes, it is valid"
5160 PRINT"No, it is not valid"
5170 PRINT
5190 INPUT"ENTER the letter opposite the correct answer":TS
5190 PRINT
5200 IF TS = "A" GOTO 5230
5210 PRINT"Wrong - the program is valid"
5220 GOTO 5250
5230 GOSUB 5590
5240 FOR I = 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":TS
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":TS
5410 GOSUB 5590
5420 FOR I = 1 TO 3
5430 PRINT I, I*I, I*I*I
5440 NEXT X
5450 PRINT
5460 PRINT"FOR I = 1 TO 3"
5470 PRINT":FOR I = 1 TO 3"
5480 PRINT"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 I = 1 TO 24
5600 PRINT
5610 NEXT I
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
5720 PRINT 2: Loop's (Intro)
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 5300
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 = i
***** Listing of Program "LESSON3" *****

5650 PRINT "DO X = X + 1"
5660 PRINT "10 PRINT X, X*X, X*X*X"
5670 PRINT "20 IF X = 9 THEN GOTO 70"
5680 PRINT "30 GOTO 20"
5690 PRINT "40 END"
6000 PRINT
6010 RETURN
6020 REM
6130 REM This is example program for COUNTER section
6140 REM
6150 PRINT
6160 PRINT "10 Z:=0"
6170 PRINT "20 Z:=Z+1"
6180 PRINT "30 IF Z = 10 THEN STOP"
6190 PRINT "40 GOTO 20"
6200 PRINT
6210 RETURN
6220 REM
6230 REM This example is for the FOR NEXT section
6240 REM
6250 PRINT
6260 PRINT "COUNTER loop"
6270 PRINT
6280 PRINT "10 FOR X = 1 TO 8"
6290 PRINT "20 PRINT X"
6300 PRINT "30 PRINT X*X"
6310 PRINT "40 IF X = 9 GOTO 30"
6320 PRINT "50 GOTO 31"
6330 PRINT "60 END"
6340 PRINT
6350 RETURN
6360 REM
6370 REM This example is for the Advanced FOR NEXT section
6380 REM
6390 REM
6400 PRINT
6410 PRINT "10 FOR X = 1 TO 2"
6420 PRINT "20 FOR Y = 1 TO 2"
6430 PRINT "30 PRINT X, Y"
6440 PRINT "40 NEXT Y"
6450 PRINT "50 NEXT X"
6460 PRINT
6470 RETURN
6480 RUN "MENU"
6490 RUN "LESSON:4"
6500 END
250  GOSUB 4090
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$ = "A"  GOTO 4750
340  IF  T$ = "C"  GOTO 4760
350  IF  T$ = "D"  GOTO 430
360  IF  T$ < "A"  GOTO 270
370  GOSUB 540
380  GOSUB 1260
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 11 through 41; 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 .1260 .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."
640  PRINT
650  INPUT"Press ENTER":T$
660  GOSUB 4090
670  PRINT"Arrays (Intro)"
680  PRINT
690  PRINT
697  PRINT"One use for ARRAYS would be to make a training program that"  
700  PRINT"listed the people on training, their time in the organization."
650 PRINT "or rank, and their training status. We could group the major categories (name, rank, training status) into three variables."  
660 PRINT "and use subscripts to provide a place for each entry in our data base."  
670 PRINT  
680 PRINT  
690 PRINT  
700 INPUT "press ENTER":T$  
710 GOSUB 4080  
720 GOSUB 4410  
730 PRINT  
740 PRINT "We could assign subscripted ARAYS to the three main categories. An ARRAY has the following format:"  
750 PRINT  
760 PRINT  
770 PRINT  
780 PRINT "The subscript is enclosed in parenthesis. Examples of valid ARRAY variables are: N$(1), R$(2), T(3)"  
790 PRINT  
800 PRINT  
810 INPUT "press ENTER":T$  
820 GOSUB 4080  
830 GOSUB 4410  
840 PRINT "We could use our ARAYS to hold the above data."  
850 PRINT "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,"  
860 PRINT "and T(0), T(1), and T(2) to represent the three training levels. Note that 0 is a valid subscript."  
870 PRINT  
880 PRINT  
890 PRINT  
900 PRINT  
910 INPUT "press ENTER":T$  
920 GOSUB 4080  
930 PRINT "0 = "CHR$(34)"John Doe"CHR$(34)  
940 PRINT "20 = "CHR$(34)"Jake Robinski"CHR$(34)  
950 PRINT "30 = "CHR$(34)"Mark Muffin"CHR$(34)  
960 PRINT "0 = "CHR$(34)"Foreman"CHR$(34)  
970 PRINT "50 = "CHR$(34)"Pean"CHR$(34)  
980 PRINT "60 = "CHR$(34)"Specialist"CHR$(34)  
990 PRINT "70 T(0) = 9"  
1000 PRINT "90 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)"  
1070 PRINT  
1080 INPUT "This program, when RUN, would print our data.....press ENTER":T$  
1090 GOSUB 4080  
1100 GOSUB 4410
1100 PRINT "This is what that program would output. Notice that we have"
1110 PRINT "used only three variables, but we added 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 "TRUE"
1220 PRINT "B FALSE"
1230 PRINT
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"
1380 PRINT
1390 PRINT "If you have a little mathematics in your background, you will"
1400 PRINT "have noticed that ARRAYS are almost the same as their math"
1410 PRINT "equivalent, except that the subscripts are in parenthesis"
1420 PRINT "instead of slightly lower and to the right of the variable."
1430 PRINT
1440 PRINT "The previous examples all dealt with a ONE-DIMENSIONAL "
1450 PRINT "ARRAY. That is, there was only one number in parenthesis"
1460 PRINT "that was significant. ARRAYS with TWO, THREE, FOUR, or more"
1470 PRINT "dimensions are possible. Most dialects of BASIC, including"
1480 PRINT "Microsoft, will handle at least 8 dimensions. An example of a"
1490 PRINT "TWO-DIMENSIONAL ARRAY would be M(2,1). 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(3,1,9) or T(5,4,3) (or any combination of numbers)"
1520 INPUT "press ENTER"; T$
1530 GOSUB 4080
1540 GOSUB 4290
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 4090
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 "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 PRINT 0 0
1720 PRINT "On the first pass, X = 1 and Y = 1. A(1,1) therefore, equals 5."
1730 PRINT "The semicolon 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$
1800 GOSUB 4080
1810 PRINT
1820 PRINT "When the second iteration of Y is done, control will pass to "
1830 PRINT "line 10 and I 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(1,1) will equal 15. Finally, Y will execute for the"
1860 PRINT "fourth time (second time) while X = 2, and A(1,Y) will have all"
1870 PRINT "numbers in the DATA statement, and the numbers will have "
1880 PRINT "been printed out."
1890 PRINT
1900 INPUT "Press ENTER": T$
1910 GOSUB 4080
1920 PRINT
1930 PRINT "Another way to look at the ARRAY is to visualize it as above."
1940 PRINT "The ARRAY has four of its pockets loaded (we ignored the 0"
1950 PRINT "pockets so the explanation would be simpler. They are still"
1960 PRINT "there, they are just not used). When A(1,Y) = 10, X must equal 1"
1970 PRINT "and Y must equal 2. Do you see?"
1980 INPUT "Press ENTER": T$
1990 GOSUB 4080
Listing or Program 'LESSON3A' ******

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"
2070 PRINT
2080 INPUT"ENTER the letter opposite the correct answer":Ts$
2090 PRINT
2100 IF Ts = "C" GOTO 2120
2110 PRINT"WRONG - the correct answer is C"
2120 PRINT"CORRECT *** OUTSTANDING ***"
2130 PRINT
2140 INPUT"press ENTER":Ts$
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":Ts$
2240 PRINT
2250 IF Ts = "A" GOTO 2280
2260 PRINT"WRONG - the correct answer is A"
2270 GOTO 2290
2280 PRINT"CORRECT "
2290 PRINT
2300 INPUT"press ENTER":Ts$
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":Ts$
2400 GOSUB 4290
2410 IF Ts = "G" GOTO 1360
2420 RETURN
2430 GOSUB 4:90
Listing of Program 'LESSONIA' ******

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 M(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"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."
2750 PRINT
2760 INPUT"Press ENTER*;T$
2770 GOSUB 4080
2780 PRINT*  Dimension and Clear*
2790 PRINT
2800 PRINT"Another problem the computer has with memory allocation, is"  
2810 PRINT"reserving enough room for STRING space. For every letter in"  
2820 PRINT"a STRING, the BASIC processor must use a little over one memory"  
2830 PRINT"location. It must be able to find the letter once it stores it"  
2840 PRINT"so it uses one location to store the letter, and another to"  
2850 PRINT"recall it where it put the STRING in the first place."
2860 PRINT"Unlike number variables, STRINGS can use up to 255 characters."
2870 PRINT"Also line numbers seldom use more than 4 - the reason is rather"  
2880 PRINT"technical. You may wish to look up how data is handled intern-
2890 PRINT"ally by the computer in a reference book. If you are going"
**** Listing of Program 'LESSONZA' ****

266: PRINT "to use more than 50 characters worth of STRING space, you must"
267: PRINT "CLEAR more room for it. The clear statement actually WIPES OUT"
268: PRINT "data space and reserves memory locations, so it must ALWAYS be"
269: PRINT "the first statement if you are going to need it."
270: PRINT
271: GOTO 4020
272: PRINT
273: INPUT "Press ENTER:"
274: GOSUB 4030
275: PRINT
276: PRINT "Dimension and Clear"
277: PRINT
278: PRINT 3000: PRINT 10 X = 150"
279: PRINT 3010: PRINT "20 CLEAR"
280: PRINT 3020: PRINT "30 PRINT T"
281: PRINT 3030: PRINT "RUN"
282: PRINT 3040: PRINT
283: PRINT 3050: PRINT "Do you see the CLEAR word zeroed out X'......press ENTER:"
284: PRINT 3060: GOSUB 4020
285: PRINT 3070: PRINT 10 CLEAR 1000"
286: PRINT 3080: PRINT "DIM A(100)"
287: PRINT 3090: PRINT "FOR X = 1 TO 100"
288: PRINT 3100: READ A(1)"
289: PRINT 3110: PRINT ".....etc."
290: PRINT
291: PRINT 3120: PRINT "The above shows the first 4 lines of a program that is going"
292: PRINT "to use more than 1000 spaces of string space (that will give"
293: PRINT "an average of 10 letters per pocket of A(1):) and is going to"
294: PRINT "use 50 pockets of the numeric array S. Note that you always"
295: PRINT "get pocket 1 for free. It is normally not used"
296: PRINT
297: PRINT 3180: PRINT "Notice the position of the CLEAR and DIM statements."
298: PRINT 3190: PRINT "What will be the output of the following program?"
299: PRINT
300: PRINT 10 A(1) = 5"
301: PRINT 20 CLEAR"
302: PRINT 30 PRINT A(1)"
303: PRINT
304: PRINT "A is 0 because the clear statement is in the wrong place."
305: PRINT
306: PRINT "the clear statement only affects STRINGS."
3330 PRINT
3340 PRINT"C Nothing because A(15) 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 - GoodoodDDDDDD 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 - GoodoodDDDDDDDDDDDDDDDDDDDD 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(1,1). 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"C0 CLEAR 1000"
3770 PRINT"C0 DIM A(2)"
3780 PRINT"C0 A(1) = 5"
**Listing of Program 'LESSON3A'**

```
1790 PRINT "A" PRINT A(2)
1800 PRINT
1810 PRINT "A The DIM statement is in the wrong place."
1820 PRINT "B The CLEAR statement is invalid."
1830 PRINT "C Nothing."
1840 PRINT
1850 INPUT "Enter the letter opposite the correct answer"; T$
1860 PRINT
1870 IF $T$ = "C" GOTO 3900
1880 PRINT "Wrong - the correct answer is C"
1890 GOTO 3910
1900 PRINT "Correct."
1910 PRINT
1920 INPUT "Press ENTER"; T$
1930 GOSUB 4290
1940 IF $T$ = "B" GOTO 2470
1950 RETURN
1960 GOSUB 4080
1970 PRINT TEST
1980 PRINT
1990 PRINT "You have completed this lesson. Enter a 'C' to continue."
2000 PRINT "The TEST or ENTER an 'R' to start over."
2010 PRINT
2020 INPUT "Enter your choice"; T$
2030 IF $T$ = "C" THEN GOTO 4740
2040 RUN
2050 REM **
2060 REM ** This subroutine clears the screen on any terminal
2070 REM **
2080 FOR I = 1 TO 24
2090 PRINT
2100 NEXT I
2110 RETURN
2120 REM **
2130 REM ** THIS IS THE MENU SUBROUTINE
2140 REM **
2150 PRINT LESSON 3B
2160 PRINT
2170 PRINT "This is the second part of a two part lesson."
2180 PRINT "It is divided into the following sections."
2190 PRINT
2200 PRINT "1) Arrays (Introduction) 2) Dimension & Clear"
2210 PRINT "3) Test"
2220 PRINT
2230 PRINT
2240 PRINT
```

149
RETURN
REM **
REM ** THIS LETS STUDENT REVIEW LESSONS AGAIN
GOSUB 4080
PRINT"Which do you wish to do?"
PRINT
PRINT"A Continue on"
PRINT"B Review this section again"
INPUT"press the letter opposite the correct answer and press ENTER":T$ IF T$ = "A" OR T$ = "B" THEN RETURN
GOTO 4290
REM
REM This subroutine is for the first Subscripts example
REM
PRINT
PRINT"Name" "Rank" "Training Status"
PRINT"John Doe" Foreman 9"
PRINT"Jake Robinski" Peon 3"
PRINT"Mark Muffin" Specialist 5"
PRINT
RETURN
REM **
REM ** THIS ROUTINE IS FOR ARRAYS EXAMPLE
REM
PRINT:"FOR Y = 1 TO 2"
PRINT:"READ A"Y","Y"
PRINT:"PRINT A"Y","Y" NEXT Y PRINT"NEXT X" PRINT"DATA 5,10,15,20" RETURN
REM **
REM ** THIS ROUTINE IS 2ND ARRAY EXAMPLE
REM
PRINT"Y"
PRINT"L. 2."
PRINT"[-----1-------]"
PRINT"1. 1 2 10 1"
PRINT"[-----]" ARRAY A"Y",11
**** Listing of Program 'LESSON2' ****

4710 PRINT: "-----[-----["
4720 PRINT
4730 RETURN
4740 RUN 'TEST2'
4750 RUN 'MENU'
4760 RUN 'LESSON2'
4770 END

07/10/62 - 11:19:48
**** Listing of Program 'TESTJ' ****

1000 REM **
1100 REM ** LESSON: TESTJ  VERSION: 1 AUG 82
1200 REM ** AUTHOR: CAPT DAN CReAGAn
1300 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
1400 REM **
1500 REM ** VARIABLES:
1600 REM **
1700 REM ** N(x) = NAMES ARrAY, USED TO READ IN SEP-
1750 REM ** LENTIAL NAMES, AND TO WRITE OUT
1800 REM **
1900 REM ** S(i) = SCORES ARrAY, USED TO READ AND
2000 REM ** WRITE SCORES
2100 REM ** Q(i) = ARrAY TO KEEP TRACK OF NUMBER OF
2200 REM ** CORRECT ANSWERS. IF AN ARRAY
2300 REM ** ELEMENT EQUALS 1, THE ANSWER WAS
2400 REM ** CORRECT
2500 REM **
2600 CLEAR 1500
2700 GO1DE 4080
2800 DIM N(1000)
2900 DIM Q(1000)
3000 DIM S(1000)
3100 PRINT " FINAL TEST 'Lesson Dl'"
3200 PRINT
3300 PRINT " This test consists of 10 questions. You must get 70 percent."
3400 PRINT " There is correct to pass. That's 7 right out of the 10 ques-"
3500 PRINT " tions. Use only capital letters in your answers, don't"
3600 PRINT " include extra spaces of letters. GOOD LUCK."
3700 PRINT
3800 INPUT ' Press ENTER to continue' TI
3900 GO1DE 498
4000 PRINT ' Which of the following is correct?'
4100 PRINT
4200 PRINT ' KILL CH1:24: SLEA1:CHAR:24:
4300 PRINT ' S1:24: KILL SLEA1:CHAR:
4400 PRINT ' V : UNSAVE CH1:24: SLEA1:CHAR:24:
4500 PRINT ' D : UNSAVE SLEA1:
4600 PRINT
4700 INPUT ' Enter the letter opposite the correct answer: '
4800 PRINT
4900 IF I = "K" GOTO 5200
5000 PRINT ' Wrong - the correct answer is K.'
5100 PRINT ' Answer 3 needs SLEA1: in quotes to be right.'
5200 PRINT ' In answers 1 & 2, SLEA1: is not a BASIC word.'
5300 PRINT ' See part 1 for details.'
5400 GOTO 4800
5500 PRINT ' CORRECT'
**** Listing of Program 'TEST' ****

1400 G 11 = 1  
1410 PRINT  
1420 INPUT 'press ENTER*TS'  
1430 GOSUB 4030  
1440 PRINT "Does the FOR NEXT combination have to be used to set up a loop"  
1450 PRINT  
1460 PRINT 'YES'  
1470 PRINT 'NO'  
1480 PRINT  
1490 INPUT 'enter the letter opposite the correct answer' ; T$  
1500 PRINT  
1510 IF T$ = "B" THEN GOTO 1530  
1520 PRINT "Wrong - the correct answer is 8"  
1530 PRINT * A loop can be made with a variety of techniques."  
1540 PRINT * including counters and IF statements, FOR - NEXT, etc."  
1550 PRINT * See part 1. FOR NEXT."  
1560 GOTO 1580  
1570 PRINT 'CORRECT'  
1580 C = 1  
1590 PRINT  
1600 INPUT 'press ENTER*TS'  
1610 GOSUB 4030  
1620 PRINT * Write out the first line of a FOR NEXT loop using i as the'  
1630 PRINT * variable. Start the loop at 1 and end it at 5. Use a step of 2."  
1640 PRINT * Use line number 20. Use all caps. Leave one space between all"  
1650 PRINT *"  
1660 PRINT  
1670 INPUT 'Your answer' ; T$  
1680 PRINT  
1690 IF T$ = "50" THEN GOTO 1770  
1700 PRINT "Wrong - the correct answer is 50 FOR I = 1 TO 5 STEP 2"  
1710 PRINT "See part 1. FOR NEXT."  
1720 GOTO 1790  
1730 PRINT 'CORRECT'  
1740 C = 1  
1750 PRINT  
1760 INPUT 'press ENTER*TS'  
1770 GOSUB 4030  
1780 PRINT 'X = 1 TO 5 STEP 2'  
1790 PRINT 'X'  
1800 PRINT 'X'  
1810 PRINT 'X'  
1820 PRINT 'X'  
1830 PRINT 'X'  
1840 PRINT 'X'  
1850 PRINT  
1860 PRINT 'Enter the output from this program. Leave one space between'  
1870 PRINT 'Terms. Make sure to consider the seed-rotation on line 200'  
1880 PRINT
**Listing of Program "TEST"**

1720 INPUT "ENTER your answer":X$  
1730 PRINT  
1740 IF X$ = "A" THEN GOTO 2200  
1750 PRINT "WRONG - the correct answer is A"  
1760 PRINT  
1770 PRINT "The first time through, a '1' is printed, then the"  
1780 PRINT "NEXT X statement increments X by 2. The second time"  
1790 PRINT "through a '3' is printed, BECAUSE the 1 because of the 'X'"  
1800 PRINT "then the NEXT X statement again"  
1810 PRINT "increments X by 2. The last time through, a 5 is printed."  
1820 PRINT "See part 1. Advanced FOR - NEXT"  
1830 GOTO 2560  
1840 PRINT "CORRECT"  
1850 X(4) = 1  
1860 PRINT  
1870 INPUT "press ENTER":X$  
1880 SUBS 4000  
1890 PRINT "How many objects (or elements) can an array have without a"  
1900 PRINT "DIM statement?"  
1910 PRINT  
1920 PRINT "10 not counting the 0 element"  
1930 PRINT "11 not counting the 0 element"  
1940 PRINT "0 not counting the 0 element"  
1950 PRINT "4 not counting the 0 element"  
1960 PRINT  
1970 INPUT "ENTER the letter opposite the correct answer":Y$  
1980 PRINT  
1990 IF Y$ = "B" THEN GOTO 2200  
2000 PRINT "WRONG - the correct answer is B"  
2010 PRINT "See part 2. ARRAYS"  
2020 GOTO 2200  
2030 PRINT "CORRECT"  
2040 X(5) = 1  
2050 PRINT  
2060 INPUT "press ENTER":X$  
2070 SUBS 4000  
2080 PRINT "How many objects are there in a 2 X 3 array?"  
2090 PRINT  
2100 INPUT "ENTER your answer":Z$  
2110 PRINT  
2120 IF Z$ = "A" THEN GOTO 3300  
2130 PRINT "WRONG - the correct answer is A"  
2140 PRINT "The easy way to count them is to multiply them."
2150 PRINT "See part 2. ARRAYS."  
2160 GOTO 3300  
2170 PRINT "CORRECT"  
2180 X(6) = 1  
2190 PRINT  
2200 END
2940 PRINT" Answer 3 has parentheses in the wrong place. C A D tr."
2950 PRINT" to use a string instead of a number for a pocket"
2960 PRINT" designation."
2970 PRINT" See part 2. ARRAYS."
2980 GOTO 2910
2990 PRINT"CORRECT" 3100 D I = 1
2910 PRINT
2920 INPUT press ENTER";I$
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." 3170 PRINT
2980 PRINT
2990 PRINT" Is the above question TRUE or FALSE?"
3000 PRINT
3010 INPUT"ENTER TRUE or FALSE";I$
3020 PRINT
3030 IF I = "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 D I = 1
3110 PRINT
3120 INPUT"press ENTER";I$
3130 GOSUB 4090
3140 FOR Y = 1 TO 10
3150 Y = Y-0:XI
3160 NEXT Y
3170 PRINT"You have finished the test, out of 10 possible correct answers"
3180 PRINT"you scored "Y"." 3190 PRINT
3200 IF 7 = THEN PRINT"YOU HAVE PASSED"
3210 GOSUB 3300
3220 IF 7 1 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 4110
DO you want your score recorded on a permanent file?

IF Ys = "Y" THEN GOTO 2670

PRINT "A YES"
PRINT "B NO"
PRINT
INPUT "Which: YS%
IF YS = "Y" THEN GOTO 2670
GOSUB 4050
PRINT "To record your score, we must open a file and put your name"
PRINT: If it. Therefore, surprisingly, we need your name. If your'
PRINT "name is not unique among the students likely to take this test,"
PRINT "please contact your test monitor for an identifying word that"
PRINT "will make you unique. Then enter that word below."
PRINT
PRINT "If you have already entered a score previously, be sure to"
PRINT "enter the same name you used before. (use all capitalsi"
PRINT
INPUT "ENTER your word or name now: YS%
OPEN "...SCOREC"
=x 0
t = x+1
IF EOF t THEN GOTO 3570
INPUT "NAME : YS%
INPUT "BIRTHDATE : YS%
IF NAME _ X THEN GOTO 3750
GOTO 3570
t = x+1
CLOSE
Y = x+1
PRINT "NAME : YS%
PRINT "BIRTHDATE : YS%
PRINT "SEX: Y%
FOR i = 1 TO x
PRINT "NAME: YS%i"
PRINT "BIRTHDATE: YS%i"
NEXT i
PRINT "You are now qualified to go to LESSON 4."
PRINT
PRINT "If you want a homework assignment, select it now."
PRINT
INPUT "Do you want to see your homework Y/N: YS%
IF YS = "Y" THEN GOTO 4140
GOTO 4150
S = i-
***** Listing of Program "TEST2" *****

07/10/83 - 01:33:05

1740 X = *1;
1770 IF EDF(J) THEN CLOSE:X=1:GOTO 3800
1780 INPUT!.N$(1), S(X)
1790 GOTO 3700
1800 OPEN"O",.I,"SCORE1"
1810 FOR M = 1 TO X
1820 PRINT$1.N$(M)
1830 PRINT$1.S$(M)
1840 NEXT M
1850 PRINT
1860 PRINT"You may now take LESSON 4. You will be returned to the MENU"
1870 PRINT"from where you may go to LESSON 4 or quit."
1880 GOTO 4120
1890 REM **
1900 REM ** THIS ROUTINE LETS STUDENT KNOW WHAT AREA HE
1910 REM ** OR SHE SHOULD REVIEW BEFORE GOING ON
1920 REM **
1930 IF J=10 THEN RETURN
1940 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
1950 PRINT
1960 IF Q(1) = 0 THEN PRINT" part 1. KILL statement"
1970 IF Q(2) = 0 OR Q(3) = 0 OR Q(4) = 0 THEN PRINT" part 1. LOOPS"
1980 IF Q(5) = 0 OR Q(6) = 0 OR Q(7) = 0 OR Q(8) = 0 THEN PRINT" part 2. ARRAYS"
1990 IF J=10 THEN PRINT" part 2. DIM statement"
2000 IF Q(10) = 0 THEN PRINT" part 2. CLEAR statement"
2010 PRINT
2020 INPUT"press ENTER";T$
2030 GOSUB 4080
2040 RETURN
2050 REM **
2060 REM ** this subroutine clears the screen"
2070 REM **
2080 FOR I = 1 TO 24
2090 PRINT
2100 NEXT I
2110 RETURN
2120 PRINT
2130 INPUT"press ENTER to return to MENU";T$
2140 RUN"MENU"
2150 PRINT
2160 INPUT"press ENTER to load your homework";T$
2170 RUN "HW2"
**** Listing of Program 'LESSON4' ****

1000 REM ** THIS PROGRAM STARTED ON 4 JUNE 1983
1010 REM ** AUTHOR: CAPTAIN DANNY J. CREEGAN
1020 REM ** TITLE: LESSON 4A
1030 REM **
1040 REM **
1050 REM **
1060 REM **
1070 GSUB 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. Creegan'
1140 PRINT 'Air Force Institute of Technology'
1150 PRINT
1160 PRINT 'OBJECTIVE: To teach the student how to make the computer communicate with standard peripheral devices.'
1170 PRINT
1180 PRINT
1190 PRINT
1200 PRINT
1210 PRINT
1220 PRINT
1230 INPUT 'press the ENTER key to continue': T$ 1
1240 CLEAR 200
1250 LL = 2: 'LL IS USED FOR HUMOR IN OPEN STATEMENT SECTION
1260 GSUB 7020
1270 GSUB 7020
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$ 1
1350 IF T$ = 'C' THEN GOTO 7490
1360 IF T$ = 'D' THEN GOTO 7470
1370 IF T$ = 'B' THEN GOTO 1510
1380 IF T$ = 'A' THEN GOTO 1290
1390 GSUB 1520
1400 GSUB 1960
1410 GSUB 2980
1420 GSUB 4630
1430 GSUB 9000
1440 GSUB 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": TS
1480 IF TS = "T" THEN GOTO 7510
1490 IF TS = "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 THEN GOTO 7470
1600 ON N GOSUB 1590, 1620, 1980, 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 (i.e. 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":$1
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
Remember, looking up the answers is not cheating, it's LEARNING.

PRINT "Remember, looking up the answers is not cheating, it's LEARNING."

PRINT
INPUT "press ENTER"; T$!
RETURN
GOSUB 7020
PRINT "LPRINT & LLIST"
1980 PRINT
1990 PRINT "LPRINT and LLIST allow you to output information to the line"
2000 PRINT "printer. They are extremely simple to use and they work."
2010 PRINT "almost exactly like the PRINT and LIST words. There are"
2020 PRINT "only a couple of things you need to keep in mind when you"
2030 PRINT "use them."
2040 PRINT
2050 PRINT "First, make sure the printer is hooked up and turned on."
2060 PRINT "and, if you are using a Cromemco, make sure the"
2070 PRINT "printer is LINKed to your terminal. Ask your"
2080 PRINT "operator how to use the LINK command."
2090 PRINT
2100 PRINT "Second, make sure there is enough paper in the printer."
2110 PRINT "to do your whole job."
2120 PRINT
2130 INPUT "press ENTER"; T$!
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 value, 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 "CHRS(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(11)"CHR$(34)"data"CHR$(34)"
2370 PRINT

181
**** Listing of Program 'LESSON4' ****

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"LLPRINT & 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"TRUE"
2680 PRINT"FALSE"
2690 PRINT
2700 INPUT"ENTER the letter opposite the correct answer and press ENTER";T$
2710 PRINT
2720 IF T$ = "S" THEN GOTO 2750
2730 PRINT"WRONG - LPRINT & 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"ALL"
2830 PRINT"ALL ALL"
2840 PRINT"CALLIST"CHR$(34)"ALL"CHR$(34)
2850 PRINT"D"LLIST"CHR$(34)"Ifn"CHR$(34)" (Ifn is your program name)"
2860 PRINT
2870 INPUT"ENTER the letter opposite the correct answer";T$
2880 PRINT
2890 IF T$="A" THEN GOTO 2970
2900 PRINT"WRONG - LLIST is just like LIST, correct answer is A"
2910 GOTO 2930
2920 PRINT"CORRECT"
2930 PRINT
2940 INPUT"press ENTER";T$
2950 GOSUB 7180
2960 IF T$ = "B" GOTO 1960
2970 RETURN
2980 GOSUB 7020
2990 PRINTSequential 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 every time 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 PRINTSerial 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 PRINTSequential 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
Listing of Program 'LESSON4' 07/10/87 - 01:56:17

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 "SEQUENTIAL access by some manuals)."
3380 PRINT
3390 INPUT "press ENTER":TS
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 "the letter opposite the correct answer":TS
3510 PRINT
3520 IF TS = "C" THEN GOSUB 7020:PRINT "Make a guess":PRINT:GOTO 7420
3530 IF TS = "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 7590
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":TS
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
**** Listing of Program 'LESSON4' ****

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 7290: PRINT 'Have you always had a drinking problem?': PRINT: PRINT 'Try again': PRINT: PRINT: GOTO 3780
3890 IF $T$ = 'B' THEN GOTO 3920
3900 PRINT 'WRONG - the correct answer is B '
3910 GOTO 3920
3920 PRINT 'CORRECT - good answer'
3930 PRINT
3940 INPUT 'press ENTER' $T$
3950 GOSUB 7020
3960 PRINT
3970 PRINT 'Sequential Files Intro'
3980 PRINT
3990 PRINT 'The statements and functions used with sequential files are:'
4000 PRINT
4010 PRINT 'OPEN INPUT$ EOi'
4020 PRINT
4030 PRINT 'CLOSE NOT$ CLOSE$
4040 PRINT
4050 PRINT 'We will be covering these words in greater detail in the'
4060 PRINT 'following sections. You don't have to memorize them now, just'
4070 PRINT 'lock them over. Generally, you would OPEN the file, either'
4080 PRINT 'INPUT your data, or PRINT it out to the disk, and then you'
4090 PRINT 'would CLOSE the file before going on.'
4100 PRINT
4110 INPUT 'press ENTER' $T$
4120 GOSUB 7020
4130 GOSUB 7290
4140 PRINT 'This is an example of how to create a SEQUENTIAL file, write'
4150 PRINT 'some data to it, and then CLOSE it. We will be examining this'
4160 PRINT 'program, and a companion program that will INPUT data from'
4170 PRINT 'the file we created, in the next sections. Generally speaking'
4180 PRINT 'line 10 OPENS the file (we'll explain more later); line 20'
4190 PRINT 'asks the operator to INPUT his/her name, and line 30 writes'
4200 PRINT 'the name out to a file called 'TEST'. Line 40 CLOSES the file.'
4210 PRINT
4220 INPUT 'press ENTER' $T$

165
GOSUB 7020
PRINT "Sequential access files are written to a disk in order, and"
PRINT "you cannot access a piece of information from the middle of the"
PRINT "file without searching through all the records for data pieces;"
PRINT "it is first to the one that has the record you want."
PRINT
PRINT "Is the above statement TRUE or FALSE?"
PRINT
PRINT "A TRUE";
PRINT "B FALSE"
PRINT "C Uhh... somewhere in-between?"
PRINT
INPUT "ENTER the letter opposite the correct answer":T$ 
PRINT
IF T$ = "A" GOTO 4410
IF T$ = "B" GOTO 4470
PRINT "WRONG - This is a key concept, please go back and"
PRINT "review this section before going on. You will be"
PRINT "given the opportunity to review in a few moments"
GOTO 4420
PRINT "CORRECT - THAT WAS IMPORTANT TO UNDERSTAND"
PRINT
INPUT "PRESS ENTER":T$ 
GOSUB 7180
IF T$ = "B" GOTO 4580
RETURN
GOSUB 7180
PRINT "OPEN statement"
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 "Also, don't OPEN a file that is already OPEN!"
4720 PRINT
4730 PRINT
4740 INPUT "press ENTER"; T$
4750 GOSUB 7020
4760 PRINT "Get the file in this manner:
4770 PRINT "OPEN is the keyword that signals the computer to expect three"
4780 PRINT "more pieces of information. The first piece is the letter 'O'"
4790 PRINT "or 'I'. 'O' stands for OUTPUT, and 'I' stands for INPUT. The'
4800 PRINT "last bit of data is the BUFFER number. In Microsoft BASIC you'
4810 PRINT "must have up to 8 buffers (more on some versions). For our pur-
4820 PRINT "poses, we will use buffer #1. The last bit of data is the'
4830 PRINT "filename. Notice that the file mode and filename are in quotes"
4840 PRINT
4850 INPUT "Type the filename opposite the correct answer"; T$
4860 PRINT
4870 IF T$ = "B" THEN GOTO 4880
4880 PRINT "Type the correct answer is B"
4890 GOTO 4870
4900 PRINT
4910 PRINT "To recap then, the format for the OPEN statement is:"
***** Listing of Program 'LESSON4' *****

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$(34)":""CHR$(34)" and "CHR$(34)"D"CHR$(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!"
5290 PRINT
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
***** Listing of Program 'LESN4' *****

5570 IF T$ = "C" THEN GOSUB 7020: LL = 1: PRINT "Starting countdown now.......": PRINT: INPUT "press ENTER for LAUNCH back to the question": T$: GOTO 5420
5580 PRINT "WRONG - the correct answer is A"
5590 GOTO 5610
5600 PRINT "CORRECT"
5610 L1 = INT
5620 INPUT "press ENTER": T$
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 7260
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 7260
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 var. as file name, 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 "OPEN" 
6090 PRINT 
6100 PRINT 
6110 PRINT "YES"
6120 PRINT
6130 INPUT "ENTER the letter opposite the correct answer": IT$
6140 PRINT
6150 IF IT$ = "B" THEN GOTO 6190
6160 PRINT "Wrong - the correct answer is B. " IT$ 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.... Courtice 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 side, if INPUT side, 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 720
6350 GOSUB 7260
6360 PRINT "If the file "TEST" exists, then line 10 will re-open it: line"
6370 PRINT "will WRITE OVER THE PREVIOUS DATA IN THE FILE, and line"
6380 PRINT "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 "ASSUMED TO THE NAME THAT WAS ENTERED IN LINE 20:"
6410 PRINT
6420 "Sequential files must be loaded into memory, manipulated and"
6430 PRINT "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
Listing or Program 'LESSON4'

440 INPUT "press ENTER:*
445 GOSUB 7620
460 PRINT "Write in the statement that will OPEN a file for OUTPUT to disk."
449 PRINT "and use buffer number 3. The filename is ""WIG"
450 PRINT
461 PRINT "Do not use a line number (although you would normally), and do"
462 PRINT "NOT put in any blanks."
463 PRINT
450 LINE INPUT "Enter your answer: ""WIG"
465 PRINT
466: GK=OPEN*"CHR$"(C4)+"+CHR$"(C4)+"+CHR$"(C4)+"+""WIG""+"CHR$"(C4)
467: IF !G=-# THEN GOTO 650
468: PRINT "Wrong - this section is difficult to visualize sometimes, how-"
469: PRINT "ever, it is very important. You must review it after-
470: PRINT this class. The correct answer is: ""G"
471: GOTO 650
472: PRINT "Fabulous - without a doubt you are a programmer"
473: PRINT
474: INPUT "press ENTER:*
475: GOSUB 7620
476: PRINT "When a sequential file is OPENed, that formerly had your ."
476: PRINT "trailing records in it, you didn't want the DATA destroyed."
477: PRINT "What must you be sure to do?"
478: PRINT
479: PRINT "Not write in the middle of the file unless it is added ""O"""
480: PRINT "ELSE INPUT all the data, manipulate it, THEN PRINT it back out"
481: PRINT "Nothing, you cannot OPEN a file that was previously created"
482: PRINT "Write only on the END of the file."
483: PRINT
484: INPUT "Enter the letter opposite the correct answer:"
485: PRINT
486: IF !G=-# THEN GOTO 651
487: PRINT "Wrong - the correct answer is E"
488: GOTO 651
489: PRINT "Correct"
491: PRINT
491: INPUT "press ENTER:*
492: IF !L = 1 THEN PRINT:PRINT "SHUTTLE LAUNCH ABORTED .... COMPUTER MALFUNCTION BLAMED:PRINT:INPUT PRESS"
493: PRINT "ENTER:*
494: GOSUB 7620
495: IF !G=-# THEN GOTO 461
496: RETURN
497: GOSUB 7620
498: RETURN
499: GOSUB 7620
**** Listing of Program 'LESSON A' ****

100 PRINT "You have completed this portion of Lesson A. 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'."
110 PRINT
120 INPUT "Enter your choice now: "
130 IF "C" = INPUT THEN GOTO 210
140 IF "R" = INPUT THEN GOTO 990
150 GOTO 110
160 REM **
170 REM ** clear screen subroutine
180 REM **
190 FOR i = 1 TO 24
200 PRINT
210 NEXT i
220 RETURN
230 REM ** This is the screen subroutine
240 PRINT "LES"n A"
250 PRINT
260 PRINT: "This is the first part of a two part lesson. It is divided into the following sections:
270 PRINT
280 PRINT: "1. Introduction " 2. Sequential Files Intro.
290 PRINT: "3. PRINT & LIST 4. OPENing files"
300 PRINT
310 PRINT
320 RETURN
330 REM ** This subroutine lets student review a section
340 GOTO 100
350 PRINT "Which do you wish to do?"
360 PRINT
370 PRINT "A. Continue on"
380 PRINT "B. Review this section again"
390 PRINT
400 INPUT "Press the letter opposite the correct answer and press ENTER":
410 IF "A" = INPUT OR "B" = INPUT THEN RETURN
420 GOTO 990
430 REM ** This subroutine is for sequential intro example
440 PRINT: OPEN "CHAR.4" "CHAR.4"
450 PRINT: "Enter your name and press ENTER":
460 PRINT: "Close "
470 RETURN
480 REM ** This subroutine is for the sequential files intro example
490 PRINT: OPEN "CHAR.4" CHAR.4. CHAR.4. "TEST":
500 PRINT: "Close "
510 RETURN
7170 PRINT "30 PRINT N$"
7180 PRINT "40 CLOSE 1"
7190 PRINT
7200 RETURN
7210 REM ** This subroutine is for the OPEN statements example **
7220 PRINT
7230 PRINT "10 PRINT A$ = "CHR$(1A) ""TEST""CHR$(1A)"
7240 PRINT "20 PRINT OPEN "CHR$(1A) ""TEST""CHR$(1A)"".1,A$"
7250 PRINT
7260 RETURN
7270 END
7280 END
I'm taking this part in its entirety.

I wish to review selected areas or take the test.

I want to return to the menu.

I want to go the the first part.

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

IF T# = "D" GOTO 5740
IF T# = "D" THEN GOTO 5770
IF T# = "B" GOTO 1250
IF T# = "A" GOTO 1040
GOSUB 1260
GOSUB 1770
GOSUB 2280
GOSUB 4650
GOSUB 5260
PRINT "You are now done with this lesson. If you wish to continue"
PRINT "to the test. ENTER a T. If you want to review. ENTER an R."
INPUT ENTER your choice?
IF T# = "T" THEN GOTO 5730
IF T# = "R" THEN RUN
GOTO 115:
GOSUB 1260
GOSUB 4650
GOSUB 5260
PRINT "Please type in the number beside the area you wish"
PRINT to review 1 through 51 and then press ENTER = ENTER a 0 to go"
PRINT "to the menu."
PRINT "What is your choice?"
IF N = 0 GOTO 5740
ON N GOSUB 1260: 1770, 2530, 4650, 5780
GOTO 1260
GOSUB 5260
PRINT "CLOSE Statement"
PRINT:
PRINT "We already stated previously that the CLOSE statement was""necessary, after you were done manipulating your files. It"
PRINT "has a few variations that are nice to know.

PRINT "CLOSE (buffer #)...
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 to continue!"
1510 L$ = "OPEN" + CHR$(4)+"="+CHR$(34)+"."+CHR$(4)+"QUESTI"+CHR$(34)
1520 GOSUB 5200
1530 PRINT "Is the following program valid (lines 20 & 30 are good)"
1540 PRINT
1550 PRINT "YES it is valid"
1560 PRINT "NO it will fail because line 50 OPENs a file already used"
1570 PRINT
1580 INPUT "ENTER the letter opposite the correct answer:10"
1590 IF IS = "A" THEN GOTO 1710
1600 PRINT "Wrong - file #2 was closed by line 40, so buffer 3 could be"
1610 PRINT "used again in line 50."
1620 GOTO 1710
1630 PRINT "CORRECT"
1640 PRINT
1650 INPUT "Press ENTER to continue!"
1660 GOSUB 5200
1670 IF IS = "E" GOTO 1300
1680 RETURN
1690 GOSUB 5200
1700 PRINT
1710 PRINT "In line 30 we PRINT to the file buffer that we OPENed in line" 1720 PRINT "the variable T$. Notice that the number to the right of " 1730 PRINT "the PRINT corresponds to the buffer number. If the buffer" 1740 PRINT "was buffer 3, then, after we OPENed the file, we would " 1750 PRINT "PRINT"PRINT$","$ in line 30"
1760 PRINT
1770 INPUT "Press ENTER to continue!"
1780 GOSUB 5200
1790 PRINT
1800 GOSUB 5460
1810 PRINT "änder items to a sequential disk file. When you first read"
### Listing of Program 'LESSON4' ###

1)20 PRINT"OPEN the file. The computer sets a pointer at the beginning.
1)30 PRINT"If you tell the computer to PRINT# something, 
1)40 PRINT"it starts writing data to the disk at the place where the 
1)50 PRINT"pointer is. At the end of the PRINT# operation, the pointer
1)60 PRINT"advances, so values are written in sequence. PRINT# writes 
1)70 PRINT"data to the disk almost exactly the way PRINT writes data to 
1)80 PRINT"the screen (or LPRINT writes to the printer)."
1)90 PRINT
2)00 INPUT"Press ENTER:"
2)10 GOSUB 5200
2)20 PRINT"PRINT" 
2)30 PRINT 
2)40 PRINT"Commas and semi-colons react the same way with PRINT# that they 
2)50 PRINT"do with PRINT statements. If you were to write the following 
2)60 PRINT"program:"
2)70 PRINT 
2)80 PRINT"PRINT10 A=10.3 
2)90 PRINT"PRINT20 B=20.2 "
3)00 PRINT 
3)10 PRINT"to a disk using PRINT#1.A,B (as opposed to PRINT#1,A,B). then" 
3)20 PRINT"we would put this on the disk: 10.3 20.2"
3)30 PRINT 
3)40 PRINT "1)50 PRINT"See the extra spaces: Those are 12 blanks that BASIC writes to" 
3)60 PRINT"the disk."
3)70 PRINT 
3)80 PRINT"INPUT"press ENTER:"
3)90 GOSUB 5200
4)00 PRINT 
4)10 PRINT"PRINT1.4,B" 
4)20 PRINT 
4)30 PRINT "PRINT" 
4)40 PRINT "PRINT"The above command puts this on disk: 10.3 20.2" 
4)50 PRINT 
4)60 PRINT "PRINT"If you use a semicolon, like this, PRINT#1.A,B then you get:" 
4)70 PRINT 
4)80 PRINT 
4)90 PRINT "PRINT"10.3 20.2" 
5)00 PRINT "PRINT" 
5)10 PRINT "PRINT"There are only three spaces between the numbers. So to save" 
5)20 PRINT"space on the disk, you may want to write to disk using semi-" 
5)30 PRINT "PRINT"colors instead of commas between your variables. Either way," 
5)40 PRINT "PRINT"will work. It's just that the commas cause 10 extra blanks to" 
5)50 PRINT"be printed out to the disk."
5)60 PRINT 
5)70 INPUT"Press ENTER:"
5)80 GOSUB 5200
5)90 PRINT "PRINT"what is wrong with the following program?"
6)00 PRINT 
6)10 PRINT "PRINT10 OPEN"CHR$(24)"I:"CHR$(24) ",I."CHR$(24)"NEWPROG"CHR$(24)"
**** Listing of Program 'LESSON4A' ****  

2380 PRINT"20 PRINTI1.3412255"  
2390 PRINT"20 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#"  
2510 GOTO 2550  
2520 PRINT"CORRECT - way to GO!"  
2530 PRINT  
2540 INPUT"press ENTER":T$  
2550 GOSUB 5200  
2560 PRINT"INPUT% EOF"  
2570 PRINT  
2580 PRINT"INPUT% is similar to the INPUT word that we learned earlier"  
2590 PRINT"only it INPUTs data from a disk that previously had data"  
2600 PRINT"printed to it."

2640 PRINT  
2650 GOSUB 5570  
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  
2720 INPUT" INPUT% inputs data from a sequential disk file and stores the"  
2730 PRINT"data in a variable. INPUT% doesn't care how data was placed on"  
2740 PRINT"the disk. It could have been put there with one PRINT#"  
2750 PRINT"or twenty PRINT# statements. WHAT MATTERS TO INPUT% IS HOW THE"  
2760 PRINT"DATA IS TERMINATED ON THE DISK, AND WHAT KIND OF DATA IT IS"  
2770 PRINT"INPUT% INFLITING."

2780 PRINT  
2790 INPUT"press ENTER":T$  
2800 GOSUB 5200  
2810 PRINT"Does the INPUT% statement check to see how the data was"  
2820 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."
2890 PRINT
2900 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
3150 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
3210 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 '1.001"
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"
Listing of Program 'LESSON4A' 07/10/83 - 02:20:28

2200 PRINT "ON THE LINE, (INCLUDING THE NUMBER!) BECAUSE A STRING INPUT DOES*"
2210 PRINT "NOT RECOGNIZE SPACES AS TERMINATORS, IT ONLY RECOGNIZES CARRIAGE*"
2220 PRINT "RETURNS AND COMMAS:"
2230 PRINT
2240 PRINT "Why is this important? Because you wrote the data to disk with"
2250 PRINT "TWO variables, and you read it back with only ONE. If you tried"
2260 PRINT "to read the number after you read in the string, you would not"
2270 PRINT "find it:"
2280 PRINT
2290 INPUT "press ENTER:"
2300 GOSUB 5200
2310 PRINT "The solution to the problem of now to write STRING data and"
2320 PRINT "NUMERIC data to disk, is to separate them with carriage returns"
2330 PRINT "or ENTERs). That way you can read them back with INPUT*"
2340 PRINT "in the same way you wrote it. For example, if you"
2350 PRINT "write a string and a number to disk, do it this way:"
2360 PRINT
2370 PRINT "..... program assumes file opened correctly....."
2380 PRINT "PRINT "DANNY, JOE"
2390 PRINT "PRINT THE STRING WITH ONE LINE and the number with another."CHR$(34)
2400 PRINT "PRINT "DANNY, JOE"
2410 PRINT
2420 PRINT "And when you read it back, use two separate statements:"n
2430 PRINT
2440 INPUT"1"
2450 PRINT"2"
2460 INPUT"3"
2470 PRINT
2480 PRINT "This will solve the problem of mixing strings and numbers on disk."
2490 PRINT
2500 INPUT "press ENTER:"
2510 GOSUB 5200
2520 GOSUB 5500
2530 PRINT "If we were to run this program, and the name that was in the"
2540 PRINT "first record in the file was "DANNY JOE", line 20 would start"
2550 PRINT "with the first byte of the first record that was not a space"
2560 PRINT "or a carriage return, and load N$ with it, it would continue"
2570 PRINT "loading N$ until it encountered either a carriage return or a"
2580 PRINT "comma. If the data contained a quoted character string, then"
2590 PRINT "all the data between the quotes would be stuffed into N$ "
2600 PRINT "unless a comma or carriage return were encountered)."
2610 PRINT
2620 INPUT "press ENTER:"
2630 GOSUB 5200
2640 GOSUB 5500
2650 PRINT "Eventually, and very quickly, the string would be loaded with"
2660 PRINT "the characters 'DANNY JOE', and line 20 would print them out."
2670 PRINT

179
"If" Listing of Program 'LESSON4A' "

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."

PRINT
"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"
GOSUB 5200
PRINT
"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:" PRINT
"If we had printed a thousand and one names into the file "TEST"", "the computer would put the EOF marker in place #1002." "When we used the EOF(buffer #) test after name #1001, control" "would pass to line 60. All the names would have been printed." "Notice that we never re-open a file if we haven't closed it." PRINT
"If we tried to re-open an already opened file, we'd get an ERROR"
PRINT
"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 !"
PRINT
"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?"
OPEN 'TEST' FOR OUTPUT AS #4
CHRN#4 = 4:"CHR#4":1,"CHR#4":"TEST"
4220 PRINT "20 IF EOF(1) THEN STOP"
4230 PRINT "IF 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":T1$ 
4330 PRINT
4340 IF T1$ = "C" THEN GOTO 4380
4350 PRINT "Wrong - The problem is that the program tries 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":T1$
4410 GOSUB 5200
4420 PRINT "What is wrong with this program?"
4430 PRINT
4440 PRINT "10 OPEN CHR$(34)"I"CHR$(34).1.I"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":T1$ 
4550 PRINT
4560 IF T1$ = "A" THEN GOTO 4590
4570 PRINT "Wrong the correct answer is A"
4580 GOTO 4600
4590 PRINT "CORRECT"
4600 PRINT
4610 INPUT "press ENTER":T1$
4620 GOSUB 5200
4630 IF T1$ = "B" GOTO 2580
4640 RETURN
4650 GOSUB 5200
4660 PRINT SUMMiARY
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 study 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.

```
1000 INPUT"press ENTER"$ " Hint:"
2000 GOSUB 5200
3000 PRINT"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."
4000 PRINT
5000 PRINT"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."
6000 PRINT
7000 PRINT"Note how the comments are inserted. Look up why this is legal in your BASIC manua. You may be surprised!"
8000 PRINT
9000 INPUT"press ENTER"
1000 GOSUB 5200
1100 PRINT"CLEAR 50:": "Clears string space"
1200 PRINT"Input filename"CHR$(4)"="CHR$(4)"IN"
1300 PRINT"Output filename"CHR$(4)"OUT"
1400 OPENCHR$(4)"IN"CHR$(4)"I"";"; "Opens the INPUT file"
1500 OPENCHR$(4)"OUT"CHR$(4)"O"";"; "Opens the OUTPUT file"
1600 IF EOF(1) THEN 1400:"; "Checks for end of file in file #1"
1700 INPUT"ENTER DATA Enter 999 to stop"CHR$(4)"D"";"; "Inputs data from file #1"
1800 PRINT"Prints data to the new file"
1900 GOTO 50:"; "Goes back for more from file #1"
2000 PRINT"Input filename"CHR$(4)"="CHR$(4)"999"CHR$(4)"999"GOTO 1400:"; "Updates new file with your data"
2100 PRINT"Goes back until line 140 closes"
```

5140 PRINT
5150 INPUT "We assume a legal input file previously existed... press ENTER": IF$
5160 GOSUB 5360
5170 IF T$="S" GOTO 4650
5180 RETURN
5190 REM ** This subroutine clears the screen on any terminal
5200 FOR I = 1 TO 24
5210 PRINT
5220 NEXT I
5230 RETURN
5240 REM ** This is the menu subroutine
5250 PRINT" LESSON 4"
5260 PRINT
5270 PRINT
5280 PRINT"This is the second part of a two part lesson. It is divided"
5290 PRINT"into the following sections:"  
5300 PRINT
5310 PRINT*1) CLOSED  
5320 PRINT*2) PRINT*  
5330 PRINT*  
5340 RETURN
5350 REM ** This subroutine lets student review a section
5360 GOSUB 5280
5370 PRINT"Which do you wish to do?"
5380 PRINT
5390 PRINT"A Continue on"  
5400 PRINT"B Review this section again"  
5410 PRINT
5420 INPUT"Press the letter opposite the correct answer and press ENTER": IF$
5430 IF T$ = "A" OR T$ = "B" THEN RETURN
5440 GOTO 5360
5450 REM ** this subroutine is for sequential intro. example
5460 PRINT"OPEN "CHR$(144)"CHR$(144)"CHR$(144)"CHR$(144)"CHR$(144)"
5470 PRINT"OPEN "CHR$(144)"CHR$(144)"CHR$(144)"CHR$(144)"
5480 PRINT"OPEN "CHR$(144)"CHR$(144)"
5490 PRINT"CLOSE 1"
5500 PRINT
5510 RETURN
5520 REM ** This subroutine is for the sequential files intro example
5530 PRINT"OPEN "CHR$(144)"CHR$(144)"CHR$(144)"
5540 PRINT"OPEN "CHR$(144)"
5550 PRINT"CLOSE 1"
5560 PRINT
5570 RETURN
5580 REM ** This subroutine is for the OPEN statements example
***** Listing of Program 'LESSON4A' *****

5600 PRINT
5610 PRINT 10 AS = "CHR$(34)"*"TEST"*CHR$(34)
5620 PRINT 20 OPEN "CHR$(34)"*"CHR$(34)"*.1.A$*
5630 PRINT
5640 RETURN
5650 PRINT 10 OPEN"CHR$(34)"*"CHR$(34)"*.1."CHR$(34)"*"TEST"*CHR$(34)
5660 PRINT 20 IF EOF(1) THEN GOTO 60
5670 PRINT 30 INPUT*.1.A$*
5680 PRINT 40 PRINT W$*
5690 PRINT 50 GOTO 20
5700 PRINT 60 PRINT"CHR$(34)"*"END OF FILE ENCOUNTERED"*CHR$(34)
5710 PRINT 70 CLOSE 1*
5720 PRINT "\ldots\ldots 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
***** Listing of Program 'TEST4' *****

1000 REM **
1010 REM ** LESSON: TEST4 
1020 REM ** VERSION: 1 AUG 83
1030 REM ** AUTHOR: CAPT DAN CREAGAN
1040 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
1050 REM ** VARIABLES:
1060 REM ** N$(X) = NAMES ARRAY, USED TO READ IN SEQUENTIAL NAMES, AND TO WRITE OUT
1070 REM ** UPDATE NAMES. 
1080 REM ** S(X) = SCORES ARRAY - USED TO READ AND WRITE SCORES
1090 REM ** Q(X) = ARRAY TO KEEP TRACK OF NUMBER OF  
1100 REM ** CORRECT ANSWERS. IF AN ARRAY ELEMENT EQUALS 1, THE ANSWER WAS 
1110 REM ** VARIABLES:
1120 REM **  
1130 DIM 3000 
1140 DIM $11000) 
1150 DIM 0(10) 
1160 DIM 5(1000) 
1170 PRINT "FINAL TEST (lesson 4)"
1180 PRINT 1 20 PRINTThis test consists of 10 questions, you must get 70 percent' of them correct to pass. (that's 7 right out of the 10 ques-
1190 PRINT"tions). Use only capital letters in your answers. Don't' 
1200 PRINT"include extra spaces or letters. When you successfully complete" 
1210 PRINT"the test, you will be given a homework assignment that will" 
1220 PRINT"bring many things together for you. GOOD LUCK""
1230 PRINT 120 INPUT'press ENTER to continue'T$
1240 PRINT 130 PRINT"When you LPRINT data you must be sure the printer is on, has" 
1250 PRINT"enough paper, and is properly connected. LPRINTing does not" 
1260 PRINT"send output to the screen." 
1270 PRINT 
1280 PRINT"Is the above paragraph TRUE or FALSE?"
1290 PRINT 130 PRINT"A TRUE" 
1310 PRINT"B FALSE" 
1320 PRINT 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 I. 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 1890
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(1) = 1
1900 PRINT
1910 INPUT "press ENTER":T$
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

Buffers are used for all data transfer between the computer and the disks. See part 1. Sequential files, and part 2. OPEN & CLOSE

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 OPEN CHR$(34)'0'CHR$(34)'I,CHR$(34)'B52DATCHR$(34)

B OPEN*CHR$(34)'5,CHR$(34)'B52DAT

C OPEN*CHR$(34)'I,CHR$(34)'CHR$(34)'B52DATCHR$(34)

D None of the above

E None of the above

See part 2. OPEN.
**** Listing of Program "TEST4" ****

2380 PRINT "CORRECT"
2390 Q(S) = 1
2400 PRINT
2410 INPUT "press ENTER": T$(0)
2420 GOSUB 4400
2430 PRINT "What is wrong with the program below (assume the END statement"
2440 PRINT "in line 20 automatically closes the files when it is executed)"
2450 PRINT
2460 PRINT 10 OPEN CHR$(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 "A Nothing"
2530 PRINT 9 "A 'File already open' ERROR will be generated"
2540 PRINT 10 The wrong file mode is used"
2550 PRINT 11 Line 20 is illegal"
2560 PRINT
2570 GOSUB 4470
2580 PRINT 1 IF T$ = "B" THEN GOTO 2650
2590 PRINT 2 "Wrong - the correct answer is B"
2600 PRINT 3 Line 50 should be 'GOTO 20' to fix the problem."
2610 PRINT 4 See part 2, OPEN and CLOSE"
2620 GOTO 2670
2630 PRINT "CORRECT"
2640 Q(S) = 1
2650 PRINT
2660 INPUT "press ENTER": T$(0)
2670 GOSUB 4400
2680 PRINT "The CLOSE word may be used without a buffer number; however."
2690 PRINT "when you do that, only the most recent file is CLOSED."
2700 PRINT
2710 PRINT "Is the above statement TRUE or FALSE?"
2720 PRINT
2730 PRINT "A TRUE"
2740 PRINT "9 FALSE"
2750 PRINT
2760 GOSUB 4470
2770 PRINT
2780 IF T$ = "B" THEN GOTO 2860
2790 PRINT "Wrong - the correct answer is B"
2800 PRINT "When the CLOSE word is used without a number, ALL"
2810 PRINT "the previously opened files are closed."
***** Listing of Program 'TEST4' *****  

2940 PRINT*  See part 2. CLOSE.*  
2950 GOTO 2880  
2960 PRINT*CORRECT*  
2970 Q(J) = 1  
2980 PRINT  
2990 INPUT'press ENTER*;T$  
3000 GOSUB 4400  
3010 PRINT*What is wrong with the following program?*  
3020 PRINT  
3030 PRINT*10 OPEN*CHR$(34)+"\CHR$(34)+.1."CHR$(34)+"TEST\CHR$(34)"  
3040 PRINT*20 PRINT\#1,10;20;30"  
3050 PRINT*30 CLOSE*  
3060 PRINT  
3070 PRINT  
3080 PRINT*A Nothing*  
3090 PRINT*B Wrong file mode is used*  
3100 PRINT*C The CLOSE statement is invalid*  
3110 PRINT*D The PRINT\#1 statement should be INPUT\#1*  
3120 PRINT  
3130 GOSUB 4470  
3140 PRINT  
3150 IF T$ = "A" THEN GOTO 3090  
3160 PRINT*WRONG - the correct answer is A*  
3170 PRINT* See part 2. OPEN and CLOSE*  
3180 GOTO 3110  
3190 PRINT*CORRECT*  
3200 Q(J) = 1  
3210 PRINT  
3220 INPUT'press ENTER*;T$  
3230 GOSUB 4400  
3240 PRINT*You have opened a file correctly, and you now want to read in*  
3250 PRINT*the data from it. Type in the command you would use.*  
3260 PRINT*Use line number 10, a space, and the command. Use buffer \#8*  
3270 PRINT*and read the data into variable AS"  
3280 PRINT  
3290 LINE INPUT*Enter the command now: *;T$  
3300 PRINT  
3310 IF T$ = "10 INPUT\#8,AS" THEN GOTO 3250  
3320 PRINT*WRONG - the correct answer is : 10 INPUT\#8,AS*  
3330 PRINT* See part 2. PRINT\#, and INPUT\#*  
3340 GOTO 3270  
3350 PRINT*CORRECT*  
3360 Q(J) = 1  
3370 PRINT  
3380 INPUT'press ENTER*;T$  
3390 GOSUB 4400
**** Listing of Program 'TEST4' ****  

3200 PRINT"What is wrong with the following program?"  
3210 PRINT  
3220 PRINT"10 IF EOF(1) THEN GOTO 40"  
3230 PRINT20 INPUTA1,A%"  
3240 PRINT30 GOTO 10"  
3250 PRINT40 CLOSE"  
3260 PRINT50 END  
3270 PRINT  
3280 PRINTA Nothing"  
3290 PRINTB The file wasn't properly opened"  
3300 PRINTC Nothing will happen with the data"  
3310 PRINTD The END statement is not needed"  
3320 PRINT  
3330 GOSUB 4470  
3340 PRINT  
3350 IF x = '9' THEN GOTO 3500  
3360 PRINT"Wrong - the correct answer is 9"  
3370 PRINT"The file should be opened before an EOF check is done."  
3380 PRINT"See part 2"  
3390 GOTO 3520  
3400 PRINT"Correct"  
3410 G(v) = 1  
3420 PRINT  
3430 INPUT"Press ENTER":Ts  
3440 GOSUB 4400  
3450 FOR Y = 1 TO 10  
3460 r = Y*(11)  
3470 NEXT X  
3480 PRINT"You have finished the test. out of 10 possible correct answers:"  
3490 PRINT"You scored "Y","  
3500 PRINT  
3510 IF Y = 1 THEN PRINT"YOU HAVE PASSED"  
3520 GOSUB 4250  
3530 IF Y = 2 THEN GOTO 3710  
3540 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"  
3550 PRINT  
3560 PRINT"YOU SHOULD RETAKE LESSON 41"  
3570 PRINT  
3580 PRINT"You will be returned to the menu."  
3590 PRINT  
3600 GOTO 4490  
3610 PRINT  
3620 PRINT"Do you want your score recorded on a permanent file?"  
3630 PRINT  
3640 PRINT"YES"  
3650 PRINT"NO"
****** Listing of Program "TEST9" ******

7260 PRINT
7270 INPUT "which letter?"
7290 IF T$ = "B" THEN GOTO 4030
7310 GOSUB 4430
7340 PRINT "To record your score, we must open a file and put your name:"
7350 PRINT "in it. Therefore, surprisingly, we need your name. If your"
7360 PRINT "name is not unique among the students already taking this test,"
7370 PRINT "please contact your test instructor for an identifying word that"
7380 PRINT "will make you unique. Then enter that word below:"
7390 PRINT
7410 PRINT "If you have already entered a score previously, be sure to"
7420 PRINT "enter the same name you used before. (Use all capital letters)"
7430 PRINT
7450 INPUT "enter your word or name now:"
7470 OPEN "G", 1, "SCORE4"
7510 X = 0
7520 IF EOF(1) THEN GOTO 3980
7530 X = X + 1
7540 INPUT #1, T$, X
7550 INPUT #1, S$(X)
7560 IF X = 1 THEN GOTO 4140
7570 GOTO 3980
7580 CLOSE
7590 X = X + 1
7600 W$(X) = T$
7610 S$(X) = "$"
7620 OPEN "G", 1, "SCORE4"
7630 FOR W = 1 TO X
7640 PRINT W$, W$(W)
7650 PRINT #1, S$(W)
7660 NEXT W
7670 PRINT
7680 GOSUB 4430
7690 PRINT "you are now qualified to go to LESSON 5."
7700 PRINT
7710 PRINT
7720 INPUT "Do you wish to have a homework assignment? (Y/N)"
7730 IF T$ = "Y" THEN GOTO 4330
7740 GOTO 3590
7750 S = W
7760 IF EOF(1) THEN CLOSE #1
7770 X = X + 1
7780 IF X > 5 THEN GOTO 4440
7790 INPUT #1, W$(X), S$(X)
7800 GOTO 3980
7810 OPEN "G", 1, "SCORE4"
7820 FOR W = 1 TO X
7830 PRINT W$, W$(W)
7840 PRINT #1, S$(W)
7850 NEXT W
7860 PRINT
***** Listing of Program 'TESTA' *****

4210 PRINT##1,9
4220 PRINT##1,9
4230 NEXT w
4240 GOTO 4290
4250 IF w=10 THEN RETURN
4260 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:" 
4270 PRINT
4280 IF a-1 = 0 THEN PRINT"  LPRINT"
4290 IF a = 0 THEN PRINT"  LLIST"
4300 IF a(j) = 0 OR q(j) = 0 THEN PRINT"  Sequential Files"
4310 IF c-3 = 0 OR q(c) = 0 THEN PRINT"  OPEN and CLOSE"
4320 IF g(b) = 0 OR q(0) = 0 THEN PRINT"  PRINT#, INPUT#, and EOF"
4330 PRINT
4340 INPUT press ENTER:
4350 GOTO 4210
4360 RETURN
470 REM **
471 REM ** this subroutine clears the screen
472 REM **
473 FOR x = 1 TO 24
474 IF x = 1 TO 24
475 NEXT x
476 RETURN
477 REM ** this subroutine is for the response section
478 REM **
479 REM "ENTER the letter opposite the correct answer:" is
480 RETURN
481 PRINT
482 INPUT press ENTER to return to MENU:
483 PRINT"MENU"
484 PRINT "CLOSE"
485 PRINT "SCORE"
486 IF EOP = THEN STOP
487 INPUT##1,9
488 PRINT##1,9
489 GOTO 4540
490 PRINT#1,9
491 PRINT#1,9
492 REM **
**** Listing of Program 'LESSON3' ****

10 REM ** THIS PROGRAM STARTED ON 1 JUN 93
20 REM ** AUTHOR: CAPTAIN DAMM J. CREAGAN
30 REM ** TITLE: LESSON 5
40 REM **
50 REM **
60 REM **
70 REM **
80 DO SUB 21000
90 PRINT LESSON: BASIC 5 VERSION: 1 AUGUST 93
100 PRINT
110 PRINT TIME REQUIRED TO COMPLETE LESSON: About one hour
120 PRINT
130 PRINT
140 PRINT AUTHOR: Capt Damm J. Creagan
150 PRINT Air Force Institute of Technology
160 PRINT
170 PRINT OBJECTIVE: To teach the student how to use SUBROUTINES
180 PRINT and LIBRARY functions.
190 PRINT
200 PRINT
210 PRINT
220 PRINT
230 PRINT
240 PRINT
250 INPUT 'Press the ENTER key to continue.'$'
260 GOSUB 31000
270 GOSUB 32000
280 PRINT 'It's taking this part in its entirety.'
290 PRINT 'I wish to review selected areas.'
300 PRINT 'I want to go to the second part.'
310 PRINT 'I want to return to the Menu.'
320 PRINT
330 INPUT 'Press either capital A, B, C, or D and then press ENTER.'$
340 IF T$ = 'A' GOTO 49000
350 IF T$ = 'B' GOTO 49000
360 IF T$ = 'C' GOTO 49000
370 IF T$ = 'D' GOTO 990
380 GOSUB 29000
390 GOSUB 31000
400 GOSUB 49000
410 GOSUB 990
420 GOSUB 990
430 GOSUB 31000
440 GOSUB 29000
450 GOSUB 49000
460 GOSUB 990
470 GOSUB 990
480 GOSUB 31000
490 PRINT 'You are now done with this part of the lesson. If you ENTER a
500 PRINT 'D' you will go on to the second part. ENTER an 'A' to'
510 PRINT 'start over. Which do you want: ' or 'N'?
520 INPUT 'Enter your choice: '
530 IF T$ = 'A' THEN GOTO 250
LESSONS 2:
S3,05: 4C11:11: .17
- PLEASE TYPE IN THE NUMBER BESIDE THE AREA YOU WISH-
10:10 PRINT "to review 11 through 51 and then press ENTER - press 0 and"
10:15 PRINT "press ENTER to return to the Menu."
10:25 PRINT
10:30 INPUT "What is your choice?"
10:31 IF N = 0 THEN GOTO 40000
10:32 GOTO 2000, 3000, 4000, 5000, 6000
10:35 GOTO 10000
10:38 GOTO 20000
10:43 PRINT
10:45 PRINT
10:50 PRINT "INTRODUCTION"
10:51 PRINT
10:52 PRINT "In the last episode of our computerized book, we assigned you a"
10:55 PRINT "homework problem that dealt with disk I/O disk input/output."
10:58 PRINT "It included most of the concepts that we have been studying."
10:61 PRINT "Now we are on the downhill part of the course. That's right."
10:64 PRINT "You are almost done with the hard parts of BASIC."
10:67 PRINT
10:70 PRINT "The last obstacle is learning about SUBROUTINES, which you will:
10:71 PRINT "learn in the first half of this lesson. After that, we will"
10:74 PRINT "review the numerous library functions (but not have to reserise)
10:77 PRINT "then, we'll just have to know how they work, and then go on"
10:80 PRINT "to lesson 3."
10:83 PRINT
10:84 PRINT
10:85 INPUT "Press ENTER if"
10:86 GOTO 2000
10:88 PRINT
10:90 PRINT "INTRODUCTION"
10:91 PRINT
10:94 PRINT "Somewhere during each lesson, we emphasize the value of doing"
10:97 PRINT "extra study besides what this course teaches you. We all learn:
10:100 PRINT "more when we DO something that we have read about. This is"
10:103 PRINT "certainly true with learning a programming language."
10:106 PRINT
10:107 PRINT "For now, though, set out your favorite BASIC manual, and curl"
10:110 PRINT "it beside your computer for another lesson in BASIC."
10:113 PRINT
10:114 INPUT "Press ENTER if"
10:115 GOTO 40000
10:118 IF N = 0 THEN GOTO 2000
10:121 RETURN
10:124 GOTO 2000
10:127 PRINT
10:128 "SUBROUTINES"
***** Listing of Program "LESSON" *****

20  PRINT
21  PRINT "We use the word 'ROUTINE' to describe the statements in the"
22  PRINT "body of a program. 'SUBROUTINE' is used to describe"
23  PRINT "a mini-program that was built, attached to the main program."
24  PRINT "and used to perform a mini-task that, for some reason,"
25  PRINT "needs special attention."
26  PRINT
27  PRINT "Usually, we use subroutines to do tasks which we perform"
28  PRINT "often in our program such as printing out a menu after"n
29  PRINT "each module of a CAI program is finished by a student."
30  PRINT "That way, we don't have to write the menu program once and call"n
31  PRINT "it when we need it."
32  PRINT
33  INPUT "Press ENTER to"
34  GOSUB 2000
35  PRINT "Why would we want to use a subroutine?"
36  PRINT
37  PRINT "To do those parts of the program that are used often"
38  PRINT "Under the normal routine?"
39  PRINT
40  INPUT "Enter the letter opposite the correct answer:"
41  PRINT
42  IF A = "A" THEN GOTO 2150
43  PRINT "WRONG - this was supposed to be an easy question to answer."
44  PRINT "Obvioulsy, we screwed up somehow. We will send you back to the"
45  PRINT "beginning of this part. Try reading between the lines a little."
46  INPUT "Press ENTER to"
47  GOTO 2000
48  PRINT "CORRECT - GOOD JOB."
49  PRINT
50  PRINT "Press ENTER to"
51  GOSUB 2000
52  PRINT "SUBRoutines"
53  PRINT
54  PRINT "Subroutines are mini-programs that we put in our main program."
55  PRINT "And when we need them, we can go to them, perform the"
56  PRINT "task they were built to do, and return to the main program."
57  PRINT
58  PRINT "Subroutines differ from other forms of program control in:"
59  PRINT "that they ALWAYS RETURN CONTROL TO THE STATEMENT THAT"
60  PRINT "FOLLOWS THE STATEMENT THAT CALLED THEM. That means you"
61  PRINT "can call a subroutine anywhere within a program, and then"
62  PRINT "control goes back to the statement that called the subroutine"
63  PRINT "after the subroutine has done its job. Control will go back to"
64  PRINT "the statement that followed the calling statement."
7240 PRINT
7250 INPUT"Press ENTER";t$:
7260 GOSUB 21000
7270 PRINT"SUBROUTINES"
7280 PRINT
7290 PRINT"If you have built a program that has to continually print"
7300 PRINT"out a prompt, asking the users if they want to review"
7310 PRINT"previous sections of the program that have run, you would"
7320 PRINT"likely use a subroutine to ask the question, return to"
7330 PRINT"the main program with the answer stored in a variable."
7340 PRINT"and branch to the right part of the program. Based on the"
7350 PRINT"answer."
7360 PRINT
7370 PRINT
7380 PRINT"The following is an example such a program."
7390 PRINT
7400 INPUT"Press ENTER";t$:
7410 GOSUB 21000
7420 PRINT
7430 PRINT"We will explore the main points of this routine in the next."
7440 PRINT"section. Press ENTER to go on";t$:
7450 PRINT
7460 GOSUB 21000
7470 PRINT"Where does control transfer when a subroutine returns to the";
7480 PRINT"main program?"
7490 PRINT
7500 PRINT"A\ The beginning of the program\"
7510 PRINT"B\ The calling statement\"
7520 PRINT"C\ The statement after the calling statement\"
7530 PRINT"D\ None of the above\"
7540 PRINT
7550 INPUT"Enter the letter opposite the correct answer";t$:
7560 PRINT
7570 IF t$ = "C" THEN GOTO 3580
7580 PRINT"Wrong - control is transferred to the statement after the ";
7590 PRINT"calling statement. The correct answer is C."
7600 GOTO 3690
7610 PRINT"Correct - User job. Now we can go on"
7620 PRINT
7630 INPUT"Press ENTER";t$:
7640 GOSUB 40000
7650 IF t$ = "B" GOTO 5000
7660 RETURN
7670 GOSUB 21000
7680 PRINT"GOSUB 1 RETURN"
4120 PRINT
4121 PRINT "The set of statements that you use to implement a subroutine"
4122 PRINT "is made of the GOSUB and RETURN words. The GOSUB word is used"
4123 PRINT "almost exactly like the GOTO statement. You put the line"
4124 PRINT "number of the start of the subroutine on the right of the GOSUB"
4125 PRINT "word. When the computer gets to it, it transfers control to"
4126 PRINT "the subroutine. When the subroutine is done, it returns to the"
4127 PRINT "main program by using the RETURN statement. You cannot GOTO"
4128 PRINT "the main program from a subroutine without risking disaster."
4129 PRINT "You should always use the RETURN statement."
4130 PRINT
4131 INPUT "Press ENTER:"
4132 GOSUB 21000
4133 "READ ALL...
4134 PRINT
4135 PRINT "If line 30 and line 90 call the subroutine, and line 10010 RETURNs"
4136 INPUT "to the appropriate statement. Press ENTER:"
4137 GOSUB 21000
4138 GOSUB 4000
4139 PRINT
4140 PRINT "If line 30 calls the subroutine, what line gets control after"
4141 PRINT "the RETURN statement .... Enter the correct line number:"
4142 PRINT
4143 IF T = "50" THEN GOTO 4260
4144 PRINT "The correct answer is line 50"
4145 GOTO 4260
4146 PRINT "You are setting GOSUBDDDddd!!"
4147 PRINT
4148 INPUT "Press ENTER:"
4149 GOSUB 21000
4150 PRINT "GOSUB & RETURN"
4151 PRINT
4152 PRINT "You can have more than one GOSUB in a program, and you can"
4153 PRINT "have more than one RETURN in a subroutine. If you have more"
4154 PRINT "than one RETURN, then the computer will return when it reaches"
4155 PRINT "the first RETURN statement it comes to. Generally speaking,"
4156 PRINT "you should try to limit the number of exits from a subroutine"
4157 PRINT "because it can get very confusing if you have RETURNs stuck"
4158 PRINT "all over the place. It is usually possible to have only one"
4159 PRINT "exit to an entire program or subroutine."
4160 PRINT
4161 INPUT "Press ENTER:"
4162 GOSUB 21000
4163 PRINT "Is the following program valid?"
4164 PRINT
4165 PRINT "'C9 INPUT CHRS(4A):" "Enter a number between 0 and 26" "CHRS(4A):"

197
**** Listing of Program 'LESSONS' ****

4450 PRINT"";50 GSUB 100";
4460 PRINT"";49 END";
4470 PRINT"";100 IF N < 10 THEN PRINT "CHR$(I4)"; "Number is less than 10"; "CHR$(I4)" ELSE GOTO 120";
4480 PRINT"";110 RETURN";
4490 PRINT"";120 IF N > 10 THEN PRINT "CHR$(I4)"; "Number is more than 10"; "CHR$(I4)" ELSE GOTO 140";
4500 PRINT"";130 RETURN";
4510 PRINT"";140 PRINT "CHR$(I4)"; "Your number is 10"; "CHR$(I4)"
4520 PRINT"";150 RETURN";
4530 PRINT
4540 PRINT""; A; "Yes, but confused by too many RETURNS";
4550 PRINT""; B; "No, the program will never get passed line 110";
4560 PRINT
4570 INPUT""; ENTER the letter opposite the correct answer:"; $;
4580 PRINT
4590 IF $ = A; THEN GOTO 4610
4600 PRINT""; WRONG - the only problem is a confusing number of RETURNS"
4610 GOTO 4620
4620 PRINT""; CORRECT"
4630 PRINT
4640 INPUT""; press ENTER:"; $;
4650 GSUB 21000
4660 PRINT""; ENTER the command to go to a subroutine that starts on line"
4670 PRINT""; "Use line 100 and put one space between all terms."
4680 PRINT
4690 INPUT""; Enter the command now:"; $;
4700 PRINT
4710 IF $ = A0000000000000; THEN GOTO 4750
4720 GOTO 4750
4730 PRINT""; CORRECT"
4740 PRINT
4750 INPUT""; press ENTER:"; $;
4760 GSUB 40000
4770 IF $ = B; THEN GOTO 4800
4780 PRINT
4790 GOTO 4800
4800 GSUB 21000
4810 PRINT""; NESTED SUBROUTINES"
4820 PRINT
4830 PRINT""; What do you think the output of the following is?"
4840 PRINT
4850 GSUB 42000
5000 PRINT
5010 INPUT""; press ENTER -or the answer:"; $;
5020 GSUB 21000
5030 PRINT""; The answer is :";
5100 GSUB 42000
**** Listing of Program 'LESSON5' **** 07/10/93 - 03:20:11

5110 PRINT
5120 PRINT"Main Program"  press ENTER";Ts
5130 PRINT"Subroutine One"
5140 PRINT"Subroutine Two"
5150 PRINT"Subroutine Two"
5160 INPUT
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";Ts
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. (i.e. 10 20 30 40)"
5370 INPUT"Enter your answer";Ts
5380 PRINT
5390 IF Ts = "10 100 20 200 100 210 30" THEN GOTO 5420
5400 PRINT "WRONG - the correct answer is:10 100 20 200 100 210 30"
5410 PRINT "That was a rough one, you may want to review this part"
5420 GOTO 5430
5430 PRINT"CORRECT - gets complicated doesn't it?"
5440 PRINT
5450 INPUT"Press ENTER";Ts
5460 GOSUB 46000
5470 IF Ts = "B" THEN GOTO 5000
5480 PRINT ON GOSUB
5490 PRINT"Remember the ON GOTO statement from our previous lessons?"
5500 PRINT
5510 PRINT"There's a command very similar to ON GOTO that can be used with"
5520 PRINT"subroutines. You have probably already guessed that the "
5530 PRINT"command is called ON GOSUB."
PRINT "Here is an example:"
PRINT
PRINT "ENTER A NUMBER BETWEEN 1 AND 4";
PRINT ON N GOSUB 300, 400, 500, 600"
PRINT
PRINT IF N = 1 then subroutine 300 would be executed. If N = 2, subroutine 400 would execute, and so on.
PRINT
INPUT "PRESS ENTER": T$:
GOSUB 21000
PRINT
PRINT "INPUT ENTER FOR AN EXAMPLE": T$
GOSUB 21000
PRINT "INPUT ENTER FOR A SQUARE TABLE": CHR$(34)
PRINT "INPUT ENTER FOR A CUBE TABLE": CHR$(34)
PRINT "INPUT ENTER": A$
PRINT ON A GOSUB 1000, 2000*
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "PRESS ENTER FOR AN EXAMPLE": T$
GOSUB 21000
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "PRESS ENTER": T$
GOSUB 21000
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
PRINT "" FOR X = 17 TO 50
PRINT X, X*X*X"
*** Listing of Program 'LESSONS' ***

07/10/83 - 03:20:11

8550 IF TS = "B" THEN GOTO 3000
8560 RETURN
20980 REM **
20990 REM ** This subroutine clears the screen on any terminal
21000 FOR X = 1 TO 24
21010 PRINT
21020 NEXT X
21030 RETURN
22000 REM **
22020 REM ** This subroutine 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
30050 PRINT" 1) Introduction"
30060 PRINT" 2) Nested Subroutines"
30070 PRINT" 3) Subroutines"
30080 PRINT" 4) ON GOSUB"
30090 PRINT
30100 PRINT
30110 RETURN
30200 REM **
30300 REM ** This subroutine gives the student a chance to review the lesson
30400 REM **
40000 GOSUB 21000
40050 PRINT"Which do you wish to do?"
40100 PRINT
40130 PRINT"A Continue on"
40140 PRINT"B Review this section again"
40150 PRINT
40200 INPUT"Press the letter opposite the correct answer and press ENTER";TS
40260 IF TS = "A" OR TS = "B" THEN RETURN
40300 GOTO 100000
40400 REM **
40490 REM **
40750 PRINT"* This subroutine is for the gosub example
40980 REM **
41000 PRINT"A CAI program is a computer assisted instruction"CHR$[4]
41010 PRINT"B program that teaches students"CHR$[4]
41020 PRINT"C GOSUB 10000 ";: "---- LOOK use sub. after every major part"
41040 PRINT"D GOSUB 10000 :""---- LOOK use sub. after every major part"
41050 PRINT"E You must ensure program doesn't goto sub by accident"
41055 PRINT*100 REM*
**** Listing of Program 'LESSONS' ****

41060 PRINT"110 REM the subroutine follows"
41063 PRINT"120 REM"
41065 PRINT'10000 INPUT'CHRS(34)"Do you want to continue or review (A=Com B=Rev)"CHRS(34)"115"
41070 PRINT'10010 RETURN"
41090 RETURN
42000 PRINT'10 PRINT'CHRS(34)"Main Program"CHRS(34)
42010 PRINT'20 GOSUB 100"
42020 PRINT'30 GOSUB 200"
42030 PRINT'40 END"
42040 PRINT'100 PRINT'CHRS(34)"Subroutine One"CHRS(34)
42045 PRINT'110 GOSUB 200"
42050 PRINT'120 RETURN"
42060 PRINT'200 PRINT'CHRS(34)"Subroutine Two"CHRS(34)
42070 PRINT'210 RETURN"
42080 RETURN
48000 RUN"MENU"
49000 RUN"lesson5a"
50000 END
PRINT "I want to go to the Menu." 
PRINT 
115 INPUT "Press either capital A, B, C, or D and then press ENTER":T$  
120 IF T$ = "A" GOTO 46000  
121 IF T$ = "B" GOTO 49000  
122 IF T$ = "C" GOTO 10000  
123 IF T$ = "D" GOTO 39000  
124 GOSUB 21000  
125 GOSUB 30000  
126 GOSUB 40000  
127 GOSUB 50000  
128 GOSUB 60000  
129 GOSUB 70000  
130 GOSUB 80000  
131 GOSUB 90000  
132 GOSUB 100000  
133 GOSUB 110000  
134 PRINT 
135 PRINT "Please type in the number beside the area you wish:" 
136 PRINT 
137 PRINT "Press ENTER to return to the Menu." 
138 PRINT 
139 INPUT "What is your choice":N  
140 IF N = 0 GOTO 48000  
141 CD ON N GOSUB 21000,22000,23000,24000,25000,26000,27000,28000,29000,21000  
142 GOTO 10000  
143 GOSUB 21000  
144 PRINT "INTRODUCTION"  
200 PRINT 
201 PRINT "In this lesson we will tackle the different categories and:"  
202 PRINT "kinds of FUNCTIONS that are available to us in Microsoft BASIC."  
203 PRINT "We will use principles already covered in previous"  
204 PRINT "sessions. If you have trouble with some of the concepts, then"  
205 PRINT "you will have to review the appropriate lesson to catch up."  
206 PRINT 
207 PRINT "However, we will not be going into great depth with our"  
208 PRINT "explanations. Many FUNCTIONS have very specific uses and it"  
209 PRINT "would not be useful for you to memorize them. The idea you"  
210 PRINT "should get from this part is that there are many FUNCTIONS"  
211 PRINT "available, and that when you need them, you should get out your"  
212 PRINT "manual and look to the specific implementation of each one."
Listing of Program 'LESSON5A'

2160 PRINT
2170 INPUT'dress ENTER':T$
2180 RETURN
3000 GOSUB 21000
3100 PRINT' Functions Overview'
3200 PRINT
3300 PRINT'Remember in the second part of Lesson 1 when we described'
3400 PRINT'FUNCTIONS for the first time? We said that we would come back'
3500 PRINT'to them in another lesson. Well, this is it!''
3600 PRINT
3700 PRINT'From that lesson we should remember that functions are prc-
3800 PRINT'written instructions that perform commonly used operations.''
3900 PRINT'You can look at functions like they were mini-subroutines.''
4000 PRINT'only you don't use GOSUB or RETURN statements to call them.''
4100 PRINT'Instead, you just use the keyword associated with the FUNCTION''
4200 PRINT'and the computer performs the appropriate operation autost-
4300 PRINT'atically. In the next sections we will study two types of'
4400 PRINT'functions, Library, and User Functions.''
4500 PRINT
4600 INPUT'dress ENTER':T$
4700 GOSUB 21000
4800 PRINT' Functions Overview'
4900 PRINT
5000 PRINT'Library functions contain useful operations that have been'
5100 PRINT'written and stored in the computer, and are there whenever you'
5200 PRINT'need them. User functions are functions that you make up'
5300 PRINT'by inserting the instruction to make them in your program.''
5400 PRINT' Then, when your program needs the special USER FUNCTION, it'
5500 PRINT'can call on it with a special word.''
5600 PRINT
5700 PRINT'There are many functions, and, depending on the specific'
5800 PRINT'implementation of Microsoft BASIC, you probably have at least'
5900 PRINT'20 [21] LIBRARY FUNCTIONS stored in your computer. On the next'
6000 PRINT'screen is a list of the typical set of LIBRARY FUNCTIONS.''
6100 PRINT
6200 INPUT'dress ENTER':T$
6300 GOSUB 21000
6400 PRINT'Typical Library Functions'
6500 PRINT
6600 GOSUB 41000
6700 PRINT
6800 PRINT'Specific examples of these functions will be given later, or'
6900 PRINT'you may look them up in your BASIC manual.''
7000 INPUT'dress ENTER':T$
7100 GOSUB 21000
7200 PRINT 'Is the following statement TRUE or FALSE?"
**Listing of Program 'LESSON5A'**

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 program."
3460 PRINT
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 in the computer.'
3630 PRINT
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 3800
3780 RETURN
3790 GOSUB 21000
3800 PRINT "Library Functions'"
3810 GOSUB 41000
3820 GOSUB 40000
3830 PRINT
3840 PRINT '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.'
3850 PRINT
3860 INPUT 'Press ENTER': T$
3870 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 logarithm 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*:
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 *
4390 PRINT*addition to this, you can use a FUNCTION as an argument for a*
4400 PRINT*FUNCTION.*
4410 PRINT
4420 INPUT*Press ENTER* + for an example*:
4430 GOSUB 21000:
4450 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 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*real part of any Library function.*
4550 PRINT
4560 INPUT*Press ENTER*:
4570 GOSUB 21000
4590 PRINT*Is the following statement valid*?
4590 PRINT
4600 PRINT*If you are not sure, try it on a calculator*
4610 PRINT
**** Listing of Program 'LESSON5A' ****

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
4555 PRINT"CORRECT"
4556 PRINT
4557 PRINT
4558 INPUT"press ENTER";T$
4559 GO SUB 21000
4560 PRINT
4561 GO SUB 43000
4562 PRINT
4563 PRINT"Library Functions"
4564 PRINT
4565 PRINT"The expression ABS(M) is evaluated first, then the outside"
4566 PRINT"expression - SQRT(M) is evaluated next. Remember from the"
4567 PRINT"first lesson when we said that the parenthesis is the highest"
4568 PRINT"priority arithmetic expression? That means that any expression"
4569 PRINT"that is within parenthesis will be evaluated first. If more"
4570 PRINT"than one set of parenthesis is used, then the expression with-"
4571 PRINT"in the inner-most set of parenthesis is evaluated first."
4572 PRINT
4573 INPUT"press ENTER";T$
4574 GO SUB 21000
4575 PRINT"Is the following sentence TRUE or FALSE?"
4576 PRINT
4577 PRINT"You can write your own library functions in special cases."
4578 PRINT
4579 PRINT"A. TRUE"
4580 PRINT"B. FALSE"
4581 PRINT
4582 INPUT"ENTER the letter opposite the correct answer";T$
4583 PRINT
4584 IF T$ = "B" THEN GOTO 4587
4585 PRINT"WRONG - library functions are permanently stored in the "
4586 PRINT"computer and cannot be created. The correct answer is B"
4587 GOTO 4590
4588 PRINT"CORRECT"
4589 PRINT
4590 INPUT"press ENTER";T$
4591 GO SUB 21000
4592 PRINT"In the next few screens, we will ask you questions concerning"
4593 PRINT"library functions. You should get out your BASIC manual and"
****** Listing or Program 'LESSON5A' ******

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 "de in: SQRT(10)."
4970 PRINT
4980 INPUT 'press ENTER';TS
4990 GOSUB 21000
5000 INPUT "what is the function that gives the natural log of 8";TS
5010 PRINT
5020 IF TS = "LOG(8)" THEN PRINT "CORRECT" ELSE PRINT "WRONG - ANSWER is LOG(8)"
5030 PRINT
5040 INPUT 'press ENTER';TS
5050 GOSUB 21000
5060 INPUT "Name the function that gives the absolute value of -7";TS
5070 PRINT
5080 IF TS = "ABS(-7)" THEN PRINT "CORRECT" ELSE PRINT "WRONG - the answer is ABS(-7)"
5090 PRINT
5100 INPUT 'press ENTER';TS
5110 GOSUB 21000
5120 INPUT "What is the function that gives the arctangent of 3";TS
5130 PRINT
5140 IF TS = "ATN(3)" THEN PRINT "CORRECT - GREAT JOB" ELSE PRINT "WRONG - the correct answer is ATN(3)"
5150 PRINT
5160 INPUT 'press ENTER';TS
5170 GOSUB 21000
5180 INPUT "What is the function that gives the sine of .5";TS
5190 PRINT
5200 IF TS = "SIN(.5)" THEN PRINT "CORRECT" ELSE PRINT "WRONG - the correct answer is SIN(.5)"
5210 PRINT
5220 INPUT 'press ENTER';TS
5230 GOSUB 21000
5240 INPUT "Name the function to give a random number between 0 & 1";TS
5250 PRINT
5260 IF TS = "RND(0)" THEN PRINT "CORRECT - good one" ELSE PRINT "WRONG - the correct answer is RND(0)"
5270 PRINT
5280 INPUT 'press ENTER';TS
5290 GOSUB 40000
5300 IF TS = "8" THEN GOTO 4000
5310 RETURN
5320 GOSUB 21000
5330 PRINT "User Functions"
5340 PRINT
5350 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 "ent. 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
6140 INPUT "Enter the letter opposite the correct answer": T$ 
6150 GOSUB 21000
6160 PRINT "Can user functions be carried over from one program to another?"
6170 PRINT
6180 PRINT "A Yes"
6190 PRINT "B No, they always have to be redefined"
6200 PRINT "C Both A & B above."
6210 PRINT
6220 IF T$ = "B" THEN GOTO b155
6230 IF T$ = "C" THEN GOSUB 21000: PRINT "Are you related to my wife?": PRINT "Please take another choice....."
6240 IF T$ = "A" THEN GOSUB 21000: PRINT "Correct";
6250 INPUT "Enter the letter opposite the correct answer": T$ 
6260 IF T$ = "B" THEN GOTO b155
6270 IF T$ = "C" THEN GOSUB 21000: PRINT "Please take another choice....."
6280 IF T$ = "A" THEN GOSUB 21000: PRINT "Correct";
6290 PRINT "User Functions"
6300 PRINT
6310 GOSUB 44000
6320 PRINT "There are two user functions in this program. They are defined:
6330 PRINT "in lines 10 and 20, and called on in lines 30, and 50. We will"
6340 PRINT "cover this program in greater detail in the next section."
6350 PRINT
6360 INPUT "Enter the letter opposite the correct answer": T$ 
6370 GOSUB 44000
6380 IF T$ = "B" THEN GOTO 6000
6390 RETURN
6400 GOSUB 21000
6410 PRINT "DEF Statement"
6420 PRINT "The DEF statement is used to define a user function and it has"
6430 PRINT "the following format:"
**** Listing of Program 'LESSON5A' ****

700 PRINT"Line 0:DEF FN=var(x) = (exp)"
701 PRINT
702 PRINT"You must never use the DEF statement in the IMMEDIATE mode."
703 PRINT"You always have to have a line number. In CP/M and Cromemco"
704 PRINT"systems, you must separate the terms DEF and FN by one space."
705 PRINT"In TRS-80s, you don't have to. The (func. name) is an valid"
706 PRINT"variable name. (name) is an optional parameter that may be"
707 PRINT"passed to the function. If it is used in the DEF statement, it"
708 PRINT"MUST be used when it is called in. (more on that later). The"
709 PRINT"term (exp) is the calculation that you wish the function to do."
710 PRINT
711 INPUT"press ENTER":T$
712 GO SUB 21000
713: GO SUB 44000
714: PRINT
715 PRINT"Get out your BASIC manual and look up DEF. It will show you"
716 PRINT"examples similar to this. Line 10 defines a function named"
717 PRINT"RC that will be set equal to the expression on the right side"
718 PRINT"of the statement. It will return a random number between 1""
719 PRINT"and 10. Note that this USER function uses a library function"
720 PRINT"as part of its definition. This is legal."
721 PRINT
722 PRINT
723 INPUT"press ENTER":T$
724 GO SUB 21000
725: IF T$ = 'A' THEN GOTO 7315
726 PRINT"NO - you CAN use a library function inside a USER function"
727: GOTO 7217
728 PRINT"CORRECT"
729 PRINT
730 PRINT
731 IF T$ = 'A' THEN GOTO 7315
732 PRINT"WRONG - you CAN use a library function inside a USER function"
733 GOTO 7217
734 PRINT"CORRECT"
735 PRINT
736 INPUT"press ENTER":T$
737 GO SUB 21000
738 GO SUB 44000
739 PRINT
740 PRINT"Line 30 calls on the function defined in line 10. When it"
741 PRINT"does, it is set equal to a random number between 1 and 10 and"
742 PRINT"then it is printed out in line 40. Line 30 calls on the"
743 PRINT"function in line 20, but it sends two values to the DEF state-"
744 PRINT"ment. It sends a (a random number) and the number 4. Ah!"
"790 PRINT": 'Valid variable can be passed to a function, even a string."
790 PRINT
791 INPUT": 'Press ENTER": $"
792 GOSUB 21000
792 GOSUB 44000
794 PRINT
795 PRINT ": A string could be passed in a function, but it would have to"
796 PRINT "be operated on legally within the DEF statement. In this case"
797 PRINT "the DEF statement in line 29 expects two numerical variables."
798 PRINT "Note that the two variables passed in line 50 do not match"
799 PRINT "the defined variables. This is also legal. You can view the"
800 PRINT "function as its own little program, it does not know the"
801 PRINT ": value of any or the variables in the outside program."
802 INPUT": 'Press ENTER": $"
803 GOSUB 21000
804 GOSUB 44000
804 PRINT
805 PRINT "The two values that are defined as A and B will be set equal"
806 PRINT "to the corresponding values of I and 4. A will equal I and B"
807 PRINT "will equal 4. The DEF statement will then use these numbers to"
808 PRINT "calculate the (exp) part of the statement, and then Y will be"
809 PRINT "made equal to this value. Finally, the value will be printed"
810 PRINT "in line 50. You may wish to copy a program like this and 
811 PRINT "examine it to see how it works."
812 INPUT": 'Press ENTER": $"
813 GOSUB 21000
814 GOSUB 44000
815 PRINT
816 PRINT "What value would be passed to B in the second DEF statement?"
817 PRINT
818 PRINT ": The value of X"
819 PRINT "Cannot tell - not enough information"
820 PRINT
821 INPUT": 'Enter the letter opposite the correct answer": $"
822 PRINT
823 IF T$ = "A" THEN GOTO 7546
824 PRINT "Wrong - the correct answer is A"
825 GOTO 7547
826 PRINT "Correct"
827 PRINT
828 INPUT": 'Press ENTER": $"
829 GOSUB 21000
830 GOSUB 44000
831 PRINT ". Here is an example of passing a string in a user function.
832 PRINT
833 PRINT
834 PRINT
835 PRINT
836 PRINT
837 PRINT
838 PRINT
839 PRINT
840 PRINT
841 PRINT
842 PRINT
843 PRINT
844 PRINT
845 PRINT
846 PRINT
847 PRINT
848 PRINT
849 PRINT
850 PRINT
**** Listing of Program 'LESSON5A' ****

7630 PRINT "In all the examples that we have shown, we have always told"
7640 PRINT "the computer when we wanted to invoke a user function by using"
7650 PRINT "the prefix "FN": vari". That is the only way to get to your"
7660 PRINT "user function."
7670 PRINT
7680 INPUT "Press ENTER to continue"
7690 GOSUB 21000
7700 GOSUB 45000
7710 PRINT
7720 PRINT "Note in the example above, that we passed a string within a"
7730 PRINT "numeric variable, but that in the DEF statement, we set the"
7740 PRINT "numeric variable equal to LEN(string). LEN(string). is an"
7750 PRINT "example of yet another type of FUNCTION that we will study in"
7760 PRINT "the next lesson. LEN(string) returns a numeric value equal to"
7770 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."
7790 PRINT "However, because it is a NUMERIC value, its type matches the"
7790 PRINT "CSF statement, and therefore it is valid."
7800 PRINT
7810 INPUT "Press ENTER to continue"
7820 GOSUB 21000
7830 PRINT "What kind of function do you create with the DEF statement?"
7840 PRINT
7850 PRINT "A USER"*
7860 PRINT "B LIBRARY"*
7870 PRINT "C EITHER USER or LIBRARY"*
7880 PRINT "D DEFINED FUNCTIONS"*
7890 PRINT
7900 INPUT "Enter the letter opposite the correct answer": I$    
7910 IF I$ = "A" THEN GOTO 7940
7920 PRINT "WRONG - only USER defined functions are created with the DEF"
7930 PRINT "statement. The correct answer is A"
7940 GOTO 7950
7950 PRINT "CORRECT! - but that was an easy one, try the next question:"*
7960 PRINT "you think you are up to it?"
7970 PRINT "we know you are - we're just trying"
7980 PRINT "to lighten things up a bit:"*
7990 PRINT
8000 INPUT "Press ENTER to continue": I$
8010 GOSUB 21000
8020 PRINT "The following statement valid":*
8030 PRINT
8040 PRINT "10 DEF FN89(A,B) = A + B"
8050 PRINT "AYes, to call on it simply put FN88 somewhere in your program"
Listing of Program 'LESSONS5A'

07/10/83 - 02:48:20

4000: GOSUB C1000
4000: 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": TS
40060 IF TS = "A" OR TS = "B" THEN RETURN
40070 GOTO 40000
40080 REM **
40090 REM ** these are for Library functions examples
40100 REM **
40110 PRINT "ABS(exp) - gives absolute value 7) LOG(exp) - gives LOG(exp)"
40120 PRINT "ATN(exp) - arctangent in radians 8) RAND10 - random numbers"
40130 PRINT "COS(exp) - returns cosine of exp 9) SIN(exp) - sign of exp"
40140 PRINT "exp) - natural exponential 10) EXP(exp) - sine of exp"
40150 PRINT "FLOOR(exp) - gives integer of exp 11) SQR(exp) - square root"
40160 PRINT "INT(exp) - gives integer of exp 12) TAN(exp) - tangent(exp)"
40170 PRINT "any appropriate numeric expression - remember, you"
40180 PRINT "can't use ANY number for some of the functions (SQR(-1) booms')"
40190 RETURN
40200 REM **
40210 REM ** the subroutine is for the library function area
40220 REM **
40230 PRINT "10 INPUT "CHS134":Tge in the number you want the square root of"CHR134":TN"
40240 PRINT "20 PRINT SQR(TN)"
40250 GOTO 10'
40260 RETURN
40270 REM **
40280 REM ** this is another example for the library functions
40290 REM **
40300 PRINT "10 INPUT "CHS134":Enter the number you want the square root of"CHR134":IN"
40310 PRINT "20 PRINT SQR ABS(IN)"
40320 GOTO 10'
40330 RETURN
40340 REM **
40350 REM ** this example is for user functions
40360 REM **
40370 PRINT "10 DEF FNH2 = INT(RAND10) * 10"
40380 PRINT "20 DEF FNW(A,B) = A: B/2 + (A - B)"
40390 PRINT "30 X = FNH2"
40400 PRINT "40 PRINT X"
40410 PRINT "50 X = FNW(1.4)"
40420 PRINT "60 PRINT X"
40430 PRINT "70 END"
**** Listing of Program 'LESSONS' ****

44060 RETURN
44970 REM **
44980 REM ** example of DEF with strings
44990 REM **
45000 PRINT "10 DEF FN1(AR) = LEN(AR)"
45010 PRINT "20 AR = "CHR$(34)"This is a function"CHR$(34)
45020 PRINT "30 X = FN1(AR)"
45030 PRINT "40 PRINT X"
45040 RETURN
47000 GOSUB 21000
47110 PRINT "Going to test number 5 - wait patiently please"
47220 RUN "TEST5"
48000 RUN "MENU"
49000 RUN "LESSONS"
50000 END
***** Listing of Program 'TEST5' *****

07/10/83 - 04:02:5a

1000 REM **
1010 REM ** LESSON: TEST5
1020 REM ** AUTHOR: CAPT DAN CREAGAN
1030 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM ** N$1(X) = NAMES ARRAY, USED TO READ IN SEQUENTIAL NAMES, AND TO WRITE OUT
1070 REM ** S1(X) = SCORES ARRAY - USED TO READ AND UPDATE NAMES.
1080 REM ** Q(1) = ARRAY TO KEEP TRACK OF NUMBER OF CORRECT ANSWERS. IF AN ARRAY ELEMENT
1090 REM ** 1 = 0, THE ANSWER WAS CORRECT
1100 REM **
1110 CLEAR 3000
1120 GOSUB 1420
1130 DIM N$(1000)
1140 DIM S(1000)
1150 REM **
1160 PRINT "FINAL TEST (Lesson 5):"
1170 PRINT
1180 PRINT "This test consists of 10 questions, you must get 70 percent 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 successfully complete the test, you can go on to the last lesson";
1190 PRINT "GOOD LUCK"
1200 PRINT
1210 INPUT "Press ENTER to continue": IT$
1220 GOSUB 1420
1230 PRINT "The order in which the following lines will be executed"
1240 PRINT "Leave one space between each line number - i.e., if the execution sequence is ten, twenty and thirty, then type in 10 20 30";
1250 PRINT
1260 PRINT "10 GOSUB 1420";
1270 PRINT "DO PRINT "CHR\&$44";DONE\&$44.
1280 PRINT "DO END"
1290 PRINT "DO RETURN"
1300 PRINT
1310 INPUT "Enter the sequence now": IT$
1320 PRINT
1330 IF IT$ = "10 1000 20 30" THEN GOTO 1470
1340 PRINT "Wrong - the correct answer is 1 3000 20 30"
1350 PRINT "See part 1. Subroutines"
1460 SOTO 1490
1470 PRINT "CORRECT"
1480 GOTO 1 = 1
1490 PRINT
1500 INPUT "Press ENTER to Continue"
1510 GOSUB 4120
1520 PRINT "What will happen when you input then number 4 to the following";
1530 PRINT "program?"
1540 PRINT
1550 PRINT "A INPUT A";
1560 PRINT "2 ON A GOSUB 1000,2000,3000"
1570 PRINT "END"
1580 PRINT "; the rest or program is not important"
1590 PRINT
1600 PRINT "A Nothing"
1610 PRINT "S Subroutine 2000 would be called"
1620 PRINT "C Subroutine 2000 would be called"
1630 PRINT "D The program would end"
1640 PRINT
1650 GOSUB 4120
1660 PRINT
1670 IF A = "D" THEN GOTO 1720
1680 PRINT "Wrong - the correct answer is D"
1690 PRINT "If you don't satisfy one of the IN GOSUB choices"
1700 PRINT "the line defaults to the line just below it."
1710 PRINT "See part 1. ON GOSUB"
1720 GOTO 1700
1730 PRINT "CORRECT"
1740 GOTO 1 = 1
1750 PRINT
1760 INPUT "Press ENTER to Continue"
1770 GOSUB 4130
1780 PRINT "How many RETURNS can you have in a subroutine?"
1790 PRINT
1800 PRINT "D One"
1810 PRINT "S As many as you want, but they should be rest to a minimum"
1820 PRINT "C One or ever. GOSUB"
1830 PRINT "S No more than the amount of space available"
1840 PRINT
1850 GOSUB 4120
1860 PRINT
1870 IF A = "S" THEN GOTO 1900
1880 PRINT "Wrong - the correct answer is S"
1890 PRINT "See part 1. Subroutines"
1900 GOTO 1800
1910 PRINT "CORRECT"
***** Listing of Program "TESTS" *****

1720 0.01 = 1
1725 PRINT
1730 INPUT "press ENTER to test"
1735 GOSUB 4100
1740 PRINT "What is wrong with this program?"
1745 PRINT
1750 INPUT "A"
1755 PRINT " GOSUB 1000"
2000 PRINT "DO END"
2010 PRINT "1000 IF A = 10 THEN RETURN"
2020 PRINT "1005 IF A = 11 THEN RETURN"
2030 PRINT "1010 GOTO 10"
2040 PRINT
2050 PRINT "4. Nothing"
2060 PRINT "5. One of the possible exits from the subroutine is incorrect"
2070 PRINT "6. There are too many RETURN statements"
2080 PRINT "7. The IF statements aren't allowed in a subroutine like this."
2090 PRINT
2100 PRINT
2110 IF A = 8 THEN GOTO 2120
2115 PRINT "The correct answer is 8"
2120 IF A > 10 THEN A = 1
2125 PRINT "That will eventually cause the computer to get stuck and bomb. See part 1 of Subroutines."
2130 GOTO 230
2135 PRINT "Correct"
2140 L = 1
2145 GOSUB 4100
2150 PRINT "For the next few questions, you should be sure you have your"
2155 PRINT "BASIC manual available so you can figure them out correctly..."
2160 PRINT "All the questions are from part 2 of the lesson, and your"
2165 PRINT "BASIC manual."
2170 PRINT
2175 INPUT "press ENTER to test"
2180 GOSUB 4100
2185 PRINT "What is the value of the following statement?"
2190 PRINT
2195 PRINT "200 DEF I"
2200 PRINT
2205 PRINT "A = 5"
2210 PRINT "The natural logarithm of 10"
**** Listing of Program "TEST." ****

200 PRINT "109"
220 PRINT
240 PRINT 420:
260 PRINT
280 IF I$ = "1A" THEN GOTO 245:
300 PRINT "WRONG - the correct answer is -":
320 GOTO 249:
340 PRINT "CORRECT"
360 C$ = "1"
380 PRINT
400 INPUT "press ENTER*"
420 Y$ = "1"
440 PRINT "What type of function would be written by you?"
460 PRINT "Use in your answer using all capital letters. Do NOT append"
480 PRINT "the word FUNCTION on the end of your answer.
500 PRINT
520 INPUT "Use in your answer now!"$
540 PRINT
560 IF I$ = "USER" THEN GOTO 260:
580 PRINT "WRONG - the correct answer is USER"
600 GOTO 260:
620 PRINT "CORRECT"
640 C$ = "1"
660 PRINT
680 INPUT "press ENTER*"
700 Y$ = "1"
720 PRINT "What is the term you would use to define a USER function?"
740 PRINT "Hint: It is a two words. DON'T include a variable. Leave a space" between the two words:
760 PRINT
780 INPUT "Enter your answer now!"$
800 PRINT
820 IF I$ = "DEF FN" THEN GOTO 260:
840 PRINT "WRONG - the correct answer is DEF FN"
860 GOTO 260:
880 PRINT "CORRECT"
900 C$ = "1"
920 PRINT
940 INPUT "press ENTER*"
960 Y$ = "1"
980 PRINT "What is the term that would complete line 10 if we wished to"
1000 PRINT "call our the USER function at line 10?"
*** Listing of Program "TESTS" ***

240 PRINT
250 INPUT "Enter your answer now!":ans
260 PRINT
270 IF ans = "FINISH" THEN GOTO 290
280 PRINT "Wrong - the correct answer would be 8966":GOTO 240
290 PRINT"CORRECT"
300 C = 1
310 PRINT
315 INPUT "Press ENTER!":SUB AIME
320 PRINT
330 PRINT "Which of the following functions is valid?":GOTO 240
340 PRINT AIME:DEF FN1(X,M) = 1 ^ M - 1 ^ X
350 PRINT B:DEF FN2(X,M) = 1 ^ M - 1 ^ X
360 PRINT C:DEF FN3(X,M) = LEN(AIME) + (""
370 PRINT D:DEF FN4(X,M) = SIN(X)
380 PRINT E:DEF FN5(X,M) = COS(X)
390 PRINT F:DEF FN6(X,M) = EXP(X)
400 PRINT
410 J = 0
420 IF ans = "C" THEN GOTO 290
430 PRINT "Wrong - the correct answer is D":
440 GOTO 240
450 PRINT"CORRECT"
460 J = 1
470 PRINT
480 INPUT "Press ENTER!":SUB AIME
490 PRINT "What is the proper statement to exit from a subroutine?":GOTO 240
500 PRINT AIME
510 PRINT "Enter your answer now!":GOTO 240
520 PRINT "If ans = "RETURN" THEN GOTO 290
530 PRINT "Wrong - the correct answer is RETURN":GOTO 240
540 PRINT "See part 1. Subroutines"
550 GOTO 240
560 PRINT"CORRECT"
570 J = 1
580 PRINT
590 INPUT "Press ENTER!":SUB AIME
600 PRINT "Defining a subroutine with"GOTO 240
610 PRINT "two inputs":SUB AIME
620 PRINT "For K = 1 TO 10":GOTO 240
630 PRINT "Y = K+0.1":GOTO 240
640 PRINT "You" K
650 PRINT "You have finished the test, out of 10 possible correct answers!"
***** Listing of Program "TESTS" *****

100: PRINT "You scored "; com
200: IF com = 9 THEN PRINT "YOU HAVE PASSED"
300: GOTO 1200
330: PRINT "YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
340: PRINT
350: PRINT "YOU SHOULD RETAKE LESSON 5"
360: PRINT
370: PRINT "You will be returned to the menu."
380: PRINT
390: GOTO 4220
400: PRINT
410: GOTO 4220
420: PRINT
430: PRINT "Do you want your score recorded on a permanent file?"
440: PRINT
450: PRINT "YES";
460: PRINT "NO";
470: PRINT
480: INPUT "Which is it?";
490: IF T$ = "5" THEN GOTO 1200
500: GOSUB 4120
510: PRINT "To record your score, we must open a file and put your name"
520: PRINT in it. Therefore, surprisingly, we need your name. If your
530: PRINT name is not unique among the students likely to take this test,
540: PRINT please contact your test monitor for an identifying word that
550: PRINT will make you unique. Then enter that word below.
560: PRINT
570: PRINT "If you have already entered a score previously, be sure to"
580: PRINT enter the same name you used before. Use all capitals.
590: PRINT
600: INPUT "Enter your word or name now: "
610: OPEN #1, "SCORES"
620: L = 
630: IF ESF = THEN GOTO 390
640: L = L + 1
650: INPUT "#";
660: INPUT "#";
670: IF W = L THEN GOTO 350
680: GOTO 320
690: CLOSE
700: I = I + 1
710: N$ I = T$;
720: S$ I = ;
730: OPEN #1, "SCORES"
740: FOR I = 1 TO 1
750: PRINT "#", N$ I;
***** Listing of Program "TESTS" *****

07/10/85 - CALICLO

10. PRINT S
10. NEXT X
10. PRINT
10. GO TO 400
10. PRINT "You are now qualified to go to LESSON 6."
10. PRINT "You may return to the MENU or receive your homework."
10. PRINT
10. INPUT "Do you want your homework assignment (Y/N)?"
10. IF T.S. = "Y" THEN GOTO 420
10. PRINT
10. GOTO 420
10. T.S. = "Y"
10. IF END = 1 THEN CLOSE: GOTO 370
10. T.S. = "Y"
10. INPUT#1, NAME(3..11)
10. GOTO 380
10. OPEN#1, "SCORES"
10. FOR # = 1 TO 4
10. PRINT#1, NAME(#)
10. PRINT#1, NAME(#)
10. NEXT #
10. PRINT
10. GOTO 380
10. IF END = 1 THEN RETURN
10. PRINT "YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:
"
10. PRINT
10. IF "Q" = "O OR "Q" = "O OR "Q" = "O OR "Q" = "O THEN PRINT "Part 1. SUBROUTINES"
10. IF "O" = "O THEN PRINT "Part 1. OR SUBS"
10. IF "Q" = "O OR "Q" = "O OR "Q" = "O THEN PRINT "Part 2. Library Functions"
10. IF "Q" = "O OR "Q" = "O THEN PRINT "Part 2. USER Functions"
10. PRINT
10. INPUT "Press ENTER to continue"
10. CLOSE#1
10. RETURN
10. REM
10. REM ** this routine clears the screen
10. REM
10. FOR I = 1 TO 25
10. PRINT
10. NEXT I
10. RETURN
10. REM
10. REM ** this subroutine is for the response section
10. REM
10. INPUT "ENTER the letter opposite the correct answer":
10. RETURN

222
***** Listing of Program 'TESTS' *****

4200 PRINT
4201 RUN"MENU"
4240 RUN"MWS"
4280 CLOSE
4300 OPEN"I...SCORES"
4370 IF EOF THEN STOP
4390 INPUT#1, A$1
4390 PRINT#A$1
4400 GOTO 4270
***** Listing of Program 'LESSON' *****

1000 REM ** THIS PROGRAM STARTED ON 12 JUNE 1981
1100 REM ** AUTHOR: CAPTAIN BARRY J. CREGAN
1200 REM ** TITLE: LESSON 3
1300 REM **
1400 REM **
1500 REM **
1600 REM **
1700 REM **
1800 REM **
1900 REM **
2000 REM **
2100 REM **
2200 REM **
2300 REM **
2400 REM **
2500 REM **
2600 REM **
2700 REM **
2800 REM **
2900 REM **
3000 REM **
3100 REM **
3200 REM **
3300 REM **
3400 REM **
3500 REM **
3600 REM **
3700 REM **
3800 REM **
3900 REM **
4000 REM **
4100 REM **
4200 REM **
4300 REM **
4400 REM **
4500 REM **
4600 REM **
4700 REM **
4800 REM **
4900 REM **
5000 REM **
5100 REM **
5200 REM **
5300 REM **
5400 REM **
5500 REM **
5600 REM **
5700 REM **
5800 REM **
5900 REM **
6000 REM **
6100 REM **
6200 REM **
6300 REM **
6400 REM **
6500 REM **
6600 REM **
6700 REM **
6800 REM **
6900 REM **
7000 REM **
7100 REM **
7200 REM **
7300 REM **
7400 REM **
7500 REM **
7600 REM **
7700 REM **
7800 REM **
7900 REM **
8000 REM **
8100 REM **
8200 REM **
8300 REM **
8400 REM **
8500 REM **
8600 REM **
8700 REM **
8800 REM **
8900 REM **
9000 REM **
9100 REM **
9200 REM **
9300 REM **
9400 REM **
9500 REM **
9600 REM **
9700 REM **
9800 REM **
9900 REM **
1000 REM **
1010 REM **
1020 REM **
1030 REM **
1040 REM **
1050 REM **
1060 REM **
1070 REM **
1080 REM **
1090 REM **
1100 REM **
1110 REM **
1120 REM **
1130 REM **
1140 REM **
1150 REM **
1160 REM **
1170 REM **
1180 REM **
1190 REM **
1200 REM **
1210 REM **
1220 REM **
1230 REM **
1240 REM **
1250 REM **
1260 REM **
1270 REM **
1280 REM **
1290 REM **
1300 REM **
1310 REM **
1320 REM **
1330 REM **
1340 REM **
1350 REM **
1360 REM **
1370 REM **
1380 REM **
1390 REM **
1400 REM **
1410 REM **
1420 REM **
1430 REM **
1440 REM **
1450 REM **
**** Listing of Program 'LESONES' ****

1400 SGSUB s550
1410 PRINT
1420 PRINT"Please type in the number beside the area you wish"
1430 PRINT"to review. (1 through 5) and then press ENTER - press 0 and"
1440 PRINT"press ENTER to return to the Menu."
1450 PRINT
1460 INPUT"What is your choice":IN
1470 IF N = 0 GOTO 7000
1480 GOSUB 1530,1550,1560,1570,1580
1490 GOTO 1440
1500 GOSUB 1590
1510 PRINT* INTRODUCTION*
1520 PRINT
1530 PRINT"This is your last lesson! Congratulations! If you have taken"
1540 PRINT"the previous five lessons, you should be feeling a little more"
1550 PRINT"comfortable with Microsoft BASIC by now."
1560 PRINT
1570 PRINT"In this lesson we will cover STRING functions first and then"
1580 PRINT"we will learn about the Microsoft Editor in the second half."
1590 PRINT
1600 PRINT"As we learned before, a BASIC string is one or more alpha-"
1610 PRINT"numeric characters that are treated as a single collection of"
1620 PRINT"data. Using the concepts in this chapter, you can perform"
1630 PRINT"many of the same types of operations on STRINGS that you"
1640 PRINT"can perform on numeric data."
1650 PRINT
1660 PRINT* INTRODUCTION*
1670 SGSUB 551
1680 PRINT
1690 PRINT"As a small review, you should remember that string data can be"
1700 PRINT"designated in two ways. You can assign your data to a string"
1710 PRINT"variable, or you can enclose the data in quotes. Here are two"
1720 PRINT"examples."
1730 PRINT
1740 PRINT"PRINT "CHR$(129)" This is one way to designate a string "CHR$(129)"
1750 PRINT"PRINT "CHR$(129)" Another way is to put it in a variable and print it "CHR$(129)"
1760 PRINT"PRINT "CHR$(129)"
1770 PRINT
1780 PRINT"Line 10 prints the string data immediately, and line 20 loads"
1790 PRINT"the variable with the data. Variable V can then be printed when"
1800 PRINT"ever we want it."
1810 PRINT
1820 PRINT* INTRODUCTION*
1830 PRINT
1840 PRINT
1850 PRINT
1860 PRINT
1870 PRINT...
**** Listing of Program 'LESSONe' ****

100 PRINT
110 PRINT PRINT "CHR$(14)"; "This is one way to designate a string"CHR$(14):
120 PRINT PRINT US = "CHR$(14)"; "And this is another"CHR$(14):
130 PRINT PRINT"="
140 PRINT PRINT "="
150 PRINT "="
160 PRINT "="
170 PRINT"RUN"
180 PRINT
190 PRINT PRINT This is one way to designate a string'
200 PRINT PRINT And this is another'
210 PRINT PRINT
220 PRINT PRINT Here is another variation of our little program. Note that'
230 PRINT "And this is another" is not printed until line 40 is executed.'
240 PRINT PRINT
250 INPUT "press ENTER":TS
260 GOSUB 610
270 GOSUB 610
280 PRINT "INTRODUCTION"
290 PRINT
300 PRINT PRINT Remember, in this lesson, as in all of our lessons, you should'
310 PRINT "have either a good BASIC manual handy, or you should have'
320 PRINT "an experienced programmer around to help you with difficult"'
330 PRINT "problems."
340 PRINT
350 PRINT PRINT In some of the answers you will need to be sure you use the'
360 PRINT "correct case (either uppercase or lowercase). So be sure to"'
370 PRINT "read all the questions carefully."
380 PRINT
390 PRINT PRINT "Get out your Manual, or programmer, now, and let's enjoy BASIC"'
400 PRINT
410 INPUT "press ENTER":TS
420 GOSUB 610
430 PRINT "STRING ASSIGNMENT"
440 PRINT
450 PRINT PRINT "As we showed you in the introduction, you assign strings to "'
460 PRINT "a variable and then you can print the variable anywhere in the"
470 PRINT "program. That makes it easier to write long program lines."
480 PRINT "Because you don't have to keep typing in the text every time"
490 PRINT "you want to use the string data."
500 PRINT
510 PRINT PRINT "You can assign data to strings using any of the statements we"
520 PRINT "used to assign numeric data to numeric variables. LET, READ,"
530 PRINT "and INPUT are all used with string assignment (LET is optional)"
540 PRINT "just as it is with numeric data."
550 PRINT
560 INPUT "press ENTER":TS
570 GOSUB 610
2380 PRINT "Are these statements legal? (Assume the program is just for"
2390 PRINT "demonstration, and that $z is blank)"
2400 PRINT
2410 PRINT "I) READ $z"
2420 PRINT "2) INPUT $z"
2430 PRINT "3) LET $z = $y"
2440 PRINT "4) "$z"$w = "CHR$(14)" NOW IS THE TIME"CHR$(14)"
2450 PRINT "5) DATA "$z"$w = "CHR$(14)" NOW IS THE TIME"CHR$(14)"
2460 PRINT
2470 PRINT "No. the LET statement in line 20 is illegal"
2480 PRINT "No. the string assignment in line 40 is illegal"
2490 PRINT "No. you cannot read data into a string (line 10 is bad)"
2500 PRINT "Yes, all statements are legal"
2510 PRINT
2520 INPUT "ENTER the letter opposite the correct answer": $s
2530 PRINT
2540 IF $s = "D" THEN GOTO 2570
2550 PRINT "All these assignments are legal"
2560 GOTO 2580
2570 PRINT "CORRECT - SUPER"
2580 PRINT
2590 $y = "D" $w = "CHR$(14)" NOW IS THE TIME "$w"
2600 INPUT "press ENTER": $s
2610 GOSUB 6510
2620 PRINT "String Assignment"
2630 PRINT
2640 GOSUB 9790
2650 PRINT "RUN"
2660 PRINT
2670 PRINT "THE BRING"
2680 PRINT "$w IS COMING"
2690 PRINT
2700 PRINT "Notice that $y was converted to $w, and all the data was"
2710 PRINT "printed out by using just $y in print statements."
2720 PRINT
2730 INPUT "press ENTER": $s
2740 GOSUB 8510
2750 PRINT "String Assignment"
2760 PRINT
2770 GOSUB 9790
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."
**** Listing of Program 'LESSON6' ****

2340 PRINT
2350 INPUT"press ENTER":IT$  
2360 GOSUB 5510
2370 PRINT"Assign NOW IS THE TIME to a string variable called NN$"  
2380 PRINT"and use line number 10 as your statement number."
2390 PRINT
2400 PRINT"Put one blank between terms."
2410 PRINT
2420 LINE INPUT"ENTER your answer now? " ; TS$  
2430 S$ = "10 NN$ = "&CHR$(34)"NOW IS THE TIME"&CHR$(34)  
2440 H$ = "10 NN$ = "&CHR$(34)"NOW IS THE TIME"&CHR$(34)
2450 PRINT
2460 IF 'S$ = S$ OR 'H$ = H$ THEN GOTO 2000  
2470 PRINT"WRONG - the correct answer is 10 NN$ = "&CHR$(34)"NOW IS THE TIME"&CHR$(34)
2480 PRINT"(you could have also answered: 10 NN$ = "&CHR$(34)"NOW IS THE TIME"&CHR$(34))"
2490 GOTO 2000  
2500 PRINT"CORRECT"
2510 PRINT
2520 INPUT"press ENTER":IT$  
2530 GOSUB 5510
2540 PRINT"You can also INPUT string data from an external file."
2550 PRINT"the following program assumes that a file named TEST was previously created on disk."
2560 PRINT
2570 PRINT
2580 PRINT"OPEN CHR$(34)"&CHR$(34)"TEST"CHR$(34)
2590 PRINT"IF EOF THEN END"
2600 PRINT"OPEN INPUT..AS"
2610 PRINT"READ PRINT AS"
2620 PRINT"5 GOTO 20"
2630 PRINT
2640 PRINT"A program such as this is used to read in your name when you"  
2650 PRINT"take your test at the end of each lesson. The original is"  
2660 PRINT"enhanced a little, but the BASIC idea is the same."  
2670 PRINT
2680 INPUT"press ENTER":IT$  
2690 GOSUB 5510
2700 IF 'S$ = "B" THEN GOTO 2220  
2710 RETURN  
2720 GOSUB 6510
2730 PRINT"String ARRays"
2740 PRINT
2750 PRINT"You can assign string data to arrays in the same way as you"  
2760 PRINT"assign numeric data to arrays. Nearly all the rules are the"  
2770 PRINT"same. The following is an example."
2780 PRINT
2790 PRINT
2800 GOSUB 3870
**** Listing of Program 'LESSONS' **** 07/10/93 - 04:14:15

220 PRINT
230 INPUT"press ENTER";T$  
240 GOSUB a510
250 GOSUB a870
260 PRINT
270 PRINT"Notice the CLEAR statement. Remember that you normally have"
280 PRINT"only 50 - 100 characters of string space available, and if you"
290 PRINT"are going to need more, you need to tell the computer. Also,"
300 PRINT"note the DIM statement - we need declare our array size if it"
310 PRINT"is over 10"
320 PRINT
330 INPUT"press ENTER";T$  
340 GOSUB a510
350 GOSUB a870
360 PRINT
370 PRINT"The variable X acts as a counter to reference the proper pocket of"
380 PRINT"the string array. The string array is referenced exactly like"
390 PRINT"the numeric array. Note that this program will only read in"
400 PRINT"the data. If you want to print it out, you will have to add"
410 PRINT"some more statements on the bottom of the program."
420 PRINT
430 INPUT"press ENTER";T$  
440 GOSUB a510
450 PRINT"What is the CLEAR statement for in BASIC?"
460 PRINT
470 PRINT"A To clear extra number space for the computer"
480 PRINT"B To clear extra string space"
490 PRINT"C To zero all number variables"
500 PRINT"D To clear the screen"
510 PRINT"E To help the programmer understand more clearly"
520 PRINT
530 INPUT"ENTER the letter opposite the correct answer";T$  
540 PRINT
550 IF T$ = "B" THEN GOTO 380
560 PRINT"WRONG - the correct answer is B"
570 PRINT"this is an important concept, you may wish to review"
580 PRINT"lesson 3 before you go to the next section."
590 GOSUB J680
600 PRINT"CORRECT"  
610 INPUT"press ENTER";T$  
620 GOSUB a510
630 PRINT"String ARRAYS"
Listing cf Program 'LESSON8' 07/10/83 - 04:14:15

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 3250
3850 PRINT"CORRECT"
3860 PRINT
3870 INPUT"press ENTER":T$
3880 GOSUB 510
3890 PRINT"How would you find out what was in the fifth pocket of the"
3900 PRINT"single dimension array A$(i)?"
3910 PRINT
3920 PRINT"A PRINT A$(i)"
3930 PRINT"B PRINT A$(5)"
3940 PRINT"C PRINT A$"
3950 PRINT"D READ A$(i)"
3960 PRINT
3970 INPUT"press the letter opposite the correct answer":T$
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$
4070 GOSUB 510
4080 PRINT
4090 GOSUB 510
4100 PRINT* Concatenation*
4110 PRINT
4120 PRINT"You may link two strings together by using the 'plus' symbol."
4130 PRINT"For example:"n
4140 PRINT
4150 PRINT"A$ = "CHR$(34)"B$ = "CHR$(34)"
4160 PRINT"B$ = "CHR$(34)"A$ = "CHR$(34)"
4170 PRINT"C$ = A$ + A$"
4180 PRINT"D$ = "CHR$(34)"E$ = "CHR$(34)"
4190 PRINT"F$ = A$ + A$"
4200 PRINT
Listing of Program 'LESSON67' 07/10/93 - 04:14:15

4210 PRINT"Somewhere"
4220 PRINT
4230 PRINT"In this case, the '+' symbol served to 'add' the two strings."
4240 PRINT"together and create another string."
4250 INPUT"Press ENTER"; T$ 
4260 GOSUB 5510
4270 PRINT"What is the output of the following program?"
4280 PRINT
4290 PRINT"A" + "FLASH"
4300 PRINT"B" + "FLASH"
4310 PRINT"C" + "DANCE"
4320 PRINT"D" + "DANCE" + "FLASH"
4330 PRINT
4340 INPUT ENTER the letter opposite the correct answer"; T$ 
4350 PRINT
4360 IF T$ = "C" THEN GOTO 4440
4370 PRINT"Wrong - the correct answer is C"
4380 GOTO 4450
4390 PRINT"Correct"
4400 PRINT
4410 GOSUB 5500 
4420 PRINT
4430 INPUT"Press ENTER"; T$ 
4440 GOSUB 5510
4450 RETURN
4460 GOSUB 5510
4470 PRINT"String Functions"
4480 PRINT
4490 PRINT"For this section you will definitely need your BASIC manual, so"
4500 PRINT"get it out now."
4510 PRINT
4520 PRINT"As with arithmetic functions, there are STRING functions."
4530 PRINT"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"
4540 PRINT"will not have to memorize them. Rather, you should understand."
4550 PRINT"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."
4560 PRINT"String functions can be used as part of USER functions."
4570 PRINT"as you saw in lesson 5."
4580 GOSUB 5510
4590 PRINT
4600 INPUT"Press ENTER for some examples of string functions"; T$ 

231
4670 GOSUB 6510
4680 PRINT" "
4690 PRINT
4700 PRINT"* LISTING OF PROGRAM 'LESSON6' *****
4710 PRINT"* 07/10/83 - 04:14:15
4720 PRINT
4730 PRINT" "
4740 PRINT
4750 PRINT"** Listing of Program 'LESSON6' ***
4760 PRINT" 07/10/83 - 04:14:15
4770 PRINT" "
4780 PRINT
4790 PRINT"* LISTING OF PROGRAM 'LESSON6' *****
4800 PRINT"* 07/10/83 - 04:14:15
4810 PRINT" "
4820 PRINT
4830 PRINT
4840 PRINT"** Listing of Program 'LESSON6' ***
4850 PRINT" 07/10/83 - 04:14:15
4860 PRINT" "
4870 PRINT
4880 PRINT
4890 PRINT"** Listing of Program 'LESSON6' ***
4900 PRINT" 07/10/83 - 04:14:15
4910 PRINT" "
4920 PRINT
4930 PRINT
4940 PRINT
4950 PRINT
4960 PRINT
4970 PRINT
4980 PRINT
4990 PRINT
5000 PRINT
5010 GOSUB 6510
5020 PRINT" String Functions"
5030 PRINT
5040 GOSUB 6580
5050 PRINT"2) PRINT ASC('A')"
5060 PRINT
5070 PRINT"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 rounded it to be "7" decimal."
5080 PRINT"2) PRINT ASC('A')"
5090 PRINT
5100 PRINT"When your computer writes data files to disk, it usually writes:
***** Listing of Program 'LESSONa' *****

510 PRINT"the ASCII code, one letter at a time. This function has"
513 PRINT"use when you are trying to convert characters to their number."
515 PRINT"equivalent.
518 PRINT
5170 INPUT"press ENTER":I$  
5180 GOSUB A51)
5190 PRINT"  String Functions"
5200 PRINT
5210 PRINT"CHR$(I$)"
5220 PRINT
5230 PRINT"CHR$(I$) returns the opposite of the ASCII string function."
5240 PRINT"It returns a character equivalent of decimal I$. Which is 'M'."
5250 PRINT
5260 INPUT"press ENTER":I$  
5270 GOSUB A51)  
5280 PRINT"What is the output of the following program?"
5290 PRINT
52910 PRINT ASC("CHR$" + "AC") + "CHR$" + "C"
5310 PRINT"Program to return the number of characters in the string."
5320 PRINT  
5330 PRINT"This function that prints the length of the string."
5340 PRINT"If the function is in parenthesis, in this case it should return 17."
5550 PRINT"press ENTER the letter opposite the correct answer":I$  
5560 PRINT
5570 IF I$ = "B" THEN GOTO 5480
5580 PRINT"Wrong - the correct answer is "O"
5590 PRINT"be sure to use your manual"
5600 SDT 5490
5610 PRINT"Program: correct - great!"
5620 PRINT
5630 INPUT"press ENTER":I$  
5640 GOSUB A51)
5650 PRINT"  String Functions"
5660 PRINT
5670 GOSUB A580
5680 PRINT"2D PRINT LEN(A$)"
5690 PRINT
5700 PRINT"LEN(string): is a function that returns the length of the string"
***** Listing of Program 'LESSON6' *****

1290 PRINT
1300 PRINT "What is the value of LEN("CHR$(A)" & &M SWIFT"CHR$(C)"?)?"
1310 PRINT
1320 PRINT 'A 11'
1330 PRINT 'B 9'
1340 PRINT 'C 1'
1350 PRINT 'D 9'
1360 PRINT
1370 INPUT 'Enter the letter opposite the correct answer: ':
1380 PRINT
1390 IF T$ = "A" THEN GOTO 5720
1400 PRINT "Wrong - the correct answer is D" : GOTO 5730
1410 PRINT "Correct"
1420 PRINT
1430 INPUT 'Press ENTER: ':
1440 GOSUB 6710
1450 PRINT "How you have to do some work for yourself. What will be the output of the following program?"
1460 PRINT
1470 GOSUB 6780
1480 PRINT "Enter left of the following program:"
1490 PRINT "PRESS ENTER:"
1500 INPUT 'Enter the letter opposite the correct answer: ':
1510 PRINT
1520 IF T$ = "A" THEN GOTO 5920
1530 PRINT "Wrong - the correct answer is A" : GOTO 5930
1540 PRINT
1550 GOSUB 8310
1560 GOSUB 6790
1570 PRINT "Enter the output of this program:"
1580 PRINT
1590 IF T$ = "achoo" THEN GOTO 6050
1600 PRINT "Wrong - the correct answer is achoo" : GOTO 6060
**** Listing of Program "LESSONS" ****

120 PRINT "CORRECT - I'm glad to see you use the book!"
120 PRINT
200 INPUT "press ENTER":T$  
230 GOSUB 510  
240 GOSUB 590  
250 PRINT "PRINT RIGHT$[:,:,4]"
260 PRINT
270 IF T$ = "a" THEN GOTO 6170  
6150 PRINT "WRONG - the correct answer is aers"
6160 GOTO 5130  
6170 PRINT "CORRECT - good job"
6180 PRINT
290 INPUT "press ENTER":T$  
320 GOSUB 510  
330 PRINT "String Functions"
340 PRINT
350 PRINT "Here's an interesting function. It's called INKEY$ and it"
360 PRINT "strokes your keyboard ONCE and if a key is depressed, it"
370 PRINT "returns the character that was pressed. Here is an example of"
380 PRINT "how to use it."
390 PRINT
400 PRINT ": IF INKEY$ = "CHR$(24)"*CHR$(24)" THEN END"  
410 PRINT "GOTO 19"
420 PRINT
430 PRINT ": IF you type this program in exactly as shown and RUN it, it"
440 PRINT "will keep running until you press the 'B' key. Try it when you"
450 PRINT "are done here."
460 PRINT
470 INPUT "press ENTER":T$  
480 GOSUB 590  
490 IF T$ = "9" THEN GOTO 4500  
500 RETURN  
510 GOSUB 510  
520 PRINT "You have finished the first part of lesson 6. If you wish to"
530 PRINT "review this part, type in 'R'. If you want to continue to the"
540 PRINT "next half, type in 'G'"
550 PRINT
560 INPUT "Enter an R or a G"$:  
570 IF T$ = "R" THEN Run  
580 IF T$ = "G" THEN GOTO 5290  
590 GOTO 400  
600 REM **
610 REM ** This subroutine clears the screen on any terminal
**** Listing of Program 'LESSON6' ****

510 FOR I = 1 TO 24
520 PRINT.
530 NEXT I
540 RETURN
550 PRINT" LESSON 6"
560 PRINT
570 PRINT"This is the first part of a two part lesson"
580 PRINT"It is divided into the following sections."
590 PRINT
600 PRINT"1) Introduction"
610 PRINT"2) Concatenation"
620 PRINT"3) String Assignment"
630 PRINT"4) String Functions"
640 PRINT
650 PRINT"5) String ARRAYS"
660 PRINT
670 PRINT"A) Continue on"
680 PRINT"B) Review this section again"
690 PRINT
700 INPUT"Press the letter opposite the correct answer and press ENTER":IS$;
710 IF IS$ = "A" THEN T$ = "B" THEN RETURN
720 GOTO 600
730 REM **
740 REM ** subroutine for string assignment example
750 REM **
760 PRINT"10 READ A$,B$"
770 PRINT"20 PRINT A$"
780 PRINT"30 A$ = B$"
790 PRINT"40 PRINT A$"
800 PRINT"50 DATA "CHR$(144),"THE GRINCH",CHR$(40),""CHR$(44)," IS COMING",CHR$(44),"
810 PRINT"60 RETURN"
820 REM **
830 REM ** subroutine for arrays example
840 REM **
850 CLEAR 2000"
860 PRINT"20 DIM A$(120)"
870 PRINT"30 I = 0"
880 PRINT"40 I = I+1"
890 INPUT"50 ENTER up to 20 strings, ENTER 'END' to stop"CHR$(44),"(A$)
900 PRINT"60 IF A$(I) = "CHR$(44),"END"CHR$(44)," THEN END"
910 PRINT"70 PRINT"80 PRINT 40"
920 RETURN
930 REM **
940 REM ** this is subroutine for STRING FUNCTIONS

276
***** Listing of Program 'LESSON6' *****

970 REM **
980 PRINT"10 AS = "CHRS(24)*'My aching fingers'CHRS(24)"
990 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
***** Listing of Program 'LESSON6A' *****

730 30SUB 21000
730 30SUB 70000
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: TS'
920 IF TS = "A" GOTO 48000
922 IF TS = "C" GOTO 49000
920 IF TS = "B" GOTO 1000
940 IF TS = "D" GOTO 890
980 GOSUB 2000
950 GOSUB 7000
960 GOSUB 4000
970 GOSUB 6000
975 GOSUB 3000
976 GOSUB 7000
978 GOSUB 8000
979 GOSUB 12000
980 GOSUB 21000
1000 GOTO 49000
1002 GOSUB 2000
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: TN'
1040 IF X = 2 GOTO 48000
1050 CN X GOSUB 2000,3000,4000,5000,6000,7000,8000,47000
1060 GOTO 1000
1070 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 rewrite a whole BASIC line just because of one'
2070 PRINT 'error. 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
**** Listing of Program 'LESSON6A' ****

11440 PRINT
1150 INPUT "Press ENTER to"
1151 GOSUB 21100
1157 PRINT "INTRODUCTION"
1160 PRINT
1170 PRINT "The object of this half will be to set you familiar with the"
1180 PRINT "editor's key commands. The lesson will not teach you all the"
1190 PRINT "commands available. However, the core of knowledge it gives"
1199 PRINT "will let you start editing BASIC programs. For some of the"
1200 PRINT "more sophisticated commands, you should refer to your Micro-
1210 PRINT "soft Manual."
1220 PRINT
1226 PRINT
1231 INPUT "Press ENTER to"
1237 RETURN
1240 GOSUB 21100
1246 PRINT "Starting"
1250 PRINT
1259 PRINT "The editor is line oriented, meaning that you operate"
1269 PRINT "on the line at a time (and not on a screen of data like you do"
1279 PRINT "with a word processor."
1289 PRINT
1297 PRINT "Throughout this lesson, we will be using one obvious line to"
1307 PRINT "illustrate all the commands. That line is listed below."
1317 PRINT
1326 PRINT "10 FOR I = 10 TO IPRINT I: NEXT I"
1331 PRINT
1336 PRINT "The colors form what is called a MULTI STATEMENT line. Each"
1346 PRINT "of a color is entered, the computer treats the data following"
1356 PRINT "it as a new line. Therefore, on the above line we have three"
1366 PRINT "statements. Obviously, there are several errors on the line."
1376 INPUT "Press ENTER to"
1387 GOSUB 21100
1397 PRINT "Starting"
1406 PRINT
1416 PRINT "10 FOR I = 10 TO IPRINT I: NEXT I"
1421 PRINT
1437 PRINT "If we wanted to EDIT this line, we would type in the word EDIT."
1447 PRINT "followed by the line number. In this case, we would type in"
1457 PRINT "EDIT 10. You may enter the EDIT in other ways. But for this"
1467 PRINT "lesson we will always use the EDIT line number syntax."
1477 PRINT "Always access the editor into the IMMEDIATE mode."
1487 PRINT
1497 PRINT "If you want to EDIT the editor after you are done editing, then"
1507 PRINT "you just press ENTER. Pressing ENTER from the EDITOR mode"
1517 PRINT "updates the line, and puts you back in IMMEDIATE mode again."
1527 PRINT


***** Listing of Program 'LESSON6A' *****  

100 INPUT 'press ENTER'
110 GOSUB 1000
120 PRINT
130 PRINT 'Starting'
140 PRINT
150 PRINT 'In the above example, we typed in the word EDIT 10, and the'
160 PRINT 'computer put us in the EDIT mode. Note that the line number'
170 PRINT 'we are editing appeared. The cursor would normally be flashing'
180 PRINT 'just to the right of the line number.'
190 PRINT
200 INPUT 'press ENTER'
210 GOSUB 1000
220 PRINT
230 PRINT 'If A = EDIT 10 THEN PRINT 'ERROR - the correct answer is EDIT 10....TRY AGAIN' PRINT 'INPUT press ENTER'
240 PRINT 'END OF PROGRAM.
250 PRINT
260'
435: PRINT
436: PRINT "For example, let's say you entered the edit mode and the cursor
437: PRINT is just on the right of the line number. Get ahead and press the
438: PRINT 'spacebar' until you get to the end of the line.
439: PRINT 'After the line is finished, press spacebar once more to go on.'
440: PRINT
441: GO = "FOR I = 1 TO 10 PRINT: NEXT I"
442: I = 1
443: PRINT "I ="
444: PRINT "I ="
445: PRINT "I ="
446: PRINT "I ="
447: PRINT "I ="
448: PRINT "I ="
449: PRINT "I ="
450: PRINT "I ="
451: PRINT "I ="
452: PRINT "I ="
453: PRINT "I ="
454: PRINT "I ="
455: PRINT "I ="
456: PRINT "I ="
457: PRINT "I ="
458: PRINT "I ="
459: PRINT "I ="
460: PRINT "I ="
461: PRINT "I ="
462: PRINT "I ="
463: PRINT "I ="
464: PRINT "I ="
465: PRINT "I ="
466: PRINT "I ="
467: PRINT "I ="
468: PRINT "I ="
469: PRINT "I ="
470: PRINT "I ="
471: PRINT "I ="
472: PRINT "I ="
473: PRINT "I ="
474: PRINT "I ="
475: PRINT "I ="
476: PRINT "I ="
477: PRINT "I ="
478: PRINT "I ="
479: PRINT "I ="
480: PRINT "I ="
481: PRINT "I ="
482: PRINT "I ="
483: PRINT "I ="
484: PRINT "I ="
485: PRINT "I ="
486: PRINT "I ="
487: PRINT "I ="
488: PRINT "I ="
489: PRINT "I ="
490: PRINT "I ="
491: PRINT "I ="
492: PRINT "I ="
493: PRINT "I ="
494: PRINT "I ="
495: PRINT "I ="
496: PRINT "I ="
497: PRINT "I ="
498: PRINT "I ="
499: PRINT "I ="
500: PRINT "I ="
501: PRINT "I ="
502: PRINT "I ="
503: PRINT "I ="
504: PRINT "I ="
505: PRINT "I ="
506: PRINT "I ="
507: PRINT "I ="
508: PRINT "I ="
509: PRINT "I ="
510: PRINT "I ="
511: PRINT "I ="
512: PRINT "I ="
513: PRINT "I ="
514: PRINT "I ="
515: PRINT "I ="
516: PRINT "I ="
517: PRINT "I ="
518: PRINT "I ="
519: PRINT "I ="
520: PRINT "I ="
521: PRINT "I ="
522: PRINT "I ="
523: PRINT "I ="
524: PRINT "I ="
525: PRINT "I ="
526: PRINT "I ="
527: PRINT "I ="
528: PRINT "I ="
529: PRINT "I ="
530: PRINT "I ="
531: PRINT "I ="
532: PRINT "I ="
533: PRINT "I ="
534: PRINT "I ="
535: PRINT "I ="
536: PRINT "I ="
537: PRINT "I ="
538: PRINT "I ="
539: PRINT "I ="
540: PRINT "I ="
541: PRINT "I ="
542: PRINT "I ="
543: PRINT "I ="
544: PRINT "I ="
545: PRINT "I ="
546: PRINT "I ="
547: PRINT "I ="
548: PRINT "I ="
549: PRINT "I ="
550: PRINT "I ="
551: PRINT "I ="
552: PRINT "I ="
553: PRINT "I ="
554: PRINT "I ="
555: PRINT "I ="
556: PRINT "I ="
557: PRINT "I ="
558: PRINT "I ="
559: PRINT "I ="
560: PRINT "I ="
561: PRINT "I ="
562: PRINT "I ="
563: PRINT "I ="
564: PRINT "I ="
565: PRINT "I ="
566: PRINT "I ="
567: PRINT "I ="
568: PRINT "I ="
569: PRINT "I ="
570: PRINT "I ="
571: PRINT "I ="
572: PRINT "I ="
573: PRINT "I ="
574: PRINT "I ="
575: PRINT "I ="
576: PRINT "I ="
577: PRINT "I ="
578: PRINT "I ="
579: PRINT "I ="
580: PRINT "I ="
581: PRINT "I ="
582: PRINT "I ="
583: PRINT "I ="
584: PRINT "I ="
585: PRINT "I ="
586: PRINT "I ="
587: PRINT "I ="
588: PRINT "I ="
589: PRINT "I ="
590: PRINT "I ="
591: PRINT "I ="
592: PRINT "I ="
593: PRINT "I ="
594: PRINT "I ="
595: PRINT "I ="
596: PRINT "I ="
597: PRINT "I ="
598: PRINT "I ="
599: PRINT "I ="
600: PRINT "I ="
***** Listing of Program 'LESSON6A' *****

441: AS = CHRS(1) IF AS = " " THEN GOTO 4410
442: IF AS = CHRS(32) THEN GOTO 4410
443: PRINT LEFT$ (AS, VAL(N$))
444: PRINT
445: PRINT "Whoa! That was quick... If you want to do it again, ENTER a 'Y'"
446: INPUT "else ENTER an 'H'"$ H$ = 1
447: IF H$ = "Y" THEN GOSUB 21000: GOTO 4255
448: IF H$ = "H" THEN GOSUB 21000: GOTO 4450
449: GOSUB 21000
450: PRINT"NSPACEBAR. nSearch"
451: PRINT
452: PRINT "If you entered a bigger number that your line length, then"
453: PRINT "the computer just defaulted to the actual line length."
454: PRINT
455: PRINT "Also, if you were actually using the editor, you could stay"
456: PRINT "in the EDIT mode, with the cursor over the nth character. After"
457: PRINT "you pressed the spacebar:"-
458: PRINT
459: PRINT "Ok, we managed to move the spacebar around a little, what if?"
460: PRINT "we wanted to find a specific character in the line, and we"-
461: PRINT "weren't quite sure exactly how far down the line it was?"
462: PRINT "In that case we could use the nSearch feature."
463: PRINT
464: INPUT"press ENTER #"
465: GOSUB 21000
466: PRINT"
467: PRINT "NSPACEBAR. nSearch"
468: PRINT "with the search feature, right after you type an EDIT line"-
469: PRINT "you can drive the cursor to any letter in the line"
470: PRINT and if the letter is not in the line, then the editor defaults"-
471: PRINT to the 3rd or the line. The editor only searches to the"
472: PRINT "right of the cursor."
473: PRINT
474: PRINT "For example, you just type in EDIT 1 and you wish to find"
475: PRINT the letter B. All you do is press 'S' and then press 1. The"-
476: PRINT 'editor recognizes upper and lower case, so be sure the case is"-
477: PRINT 'right. So ahead and do it now. You have search for an"
478: PRINT 'letter. Just type 3 first, then type the letter "
479: PRINT
480: PRINT"10"
481: PRINT "EDIT 1"
482: PRINT"
483: AS = CHRS(32)
484: IF AS = "A" THEN AS = "a"
485: IF AS = "B" THEN AS = "b"
486: IF AS = "C" THEN AS = "c"
487: AS = 100
488: IF AS = "a" THEN GOSUB 4970
**** Listing of Program 'LESSON6A' ****

4820 N = INST:0,N,R:
4830 IF N = 0 THEN N = -1:
4870 PRINT 'LEFT#:0,N#:1':
4880 PRINT
4890 PRINT 'Now... Another quick one... ENTER 'Y' to do again, else ENTER an''
4900 INPUT 'N''':I:
4910 PRINT
4920 IF I = "Y" THEN PRINT 'Search for a '':1:GOTO 4770
4930 IF I = "N" THEN GOTO 4890
4940 GOSUB 21000:
4950 PRINT 'nSPACESAR, nSearch)'
4960 PRINT
4970 PRINT 'You will have noticed that the cursor stopped BEFORE the ''
4980 PR 'character that you were searching for. That is what is ''
4990 PR 'happened. In addition, if you asked to search for a''
5000 PR 'non-existent character, the computer printed the whole line. ''
5010 PRINT
5020 PRINT 'No doubt you will also have noticed that there is a lower case''
5030 PRINT 'n in the title nSearchi. It simply means that you can''
5040 PRINT 'search for the nth occurrence of the specific character.'
5050 PRINT For example, if we were in the EDIT mode for Line 10, and we''
5060 PRINT wanted to find the second occurrence of the letter 'R' then we''
5070 PRINT would type '2SR' and the cursor would skip over to the R in''
5080 PRINT 'PRINT '...''
5090 PRINT
5100 INPUT 'Press ENTER to: '
5110 PRINT 'nSPACESAR, nSearchi''
5120 PRINT
5130 PRINT 'Here is an example of the nSearchi feature. ''
5140 PRINT
5150 PRINT 'Enter 10 '':
5160 PRINT 'EDIT 4'':
5170 PRINT 'Find '''':
5180 PRINT 'New type 2SR to find 2nd occurrence of R in 10'':
5190 PRINT
5200 PRINT
5210 PRINT 'The cursor would stop just before the nth occurrence of R.'
5220 PRINT
5230 INPUT 'Press ENTER to: '
5240 GOSUB 21000:
5250 PRINT 'What would you type to find the 2nd occurrence of the character' '
5260 PRINT 'in a line you were editing? Assume you are already in the' '
5270 PRINT 'EDIT mode.'
5280 PRINT
5290 PRINT

141
***** Listing of Program 'LESSON9A' *****

5230 INPUT "ENTER your answer now: ":$  
5240 PRINT  
5250 IF $ = "CST" THEN GOTO 5330  
5260 PRINT "WRONG - you should type CST"  
5270 GOTO 5340  
5280 PRINT "CORRECT - GREAT!"  
5290 PRINT  
5300 INPUT "press ENTER":$  
5310 GOSUB 40000  
5320 IF $ = "B" THEN GOTO 40000  
5330 RETURN  
5340 GOSUB 21900  
5350 GOSUB 21900  
5360 PRINT "Hopefull you are catching on to the way the boys and girls at"  
5370 PRINT "Microsoft are doing things, and won't have too much trouble"  
5380 PRINT "with this command. It does what it looks like it does. It"  
5390 PRINT "deletes characters. one at a time, or 'n' at a time."  
5400 PRINT  
5410 PRINT "For example, if you are in the EDIT mode for line 10 and you"  
5420 PRINT "want to delete the 'NEW' character, then just press D and the"  
5430 PRINT "character will be enclosed in exclamation marks. The excla-"  
5440 PRINT "mation marks indicate that if you don't change things, then the"  
5450 PRINT "new line will not have the character in it."  
5460 PRINT  
5470 PRINT  
5480 INPUT "press ENTER":$  
5490 GOSUB 21900  
5500 PRINT "An example of the Delete: command would be:"

5510 PRINT  
5520 PRINT "For example, if you are in the EDIT mode for line 10 and you"  
5530 PRINT "want to delete the 'S*' character, then just press D and the"  
5540 PRINT "character will be enclosed in exclamation marks. The excla-"  
5550 PRINT "mation marks indicate that if you don't change things, then the"  
5560 PRINT "new line will not have the character in it."  
5570 PRINT  
5580 PRINT  
5590 INPUT "press ENTER":$  
5600 GOSUB 21900  
5610 PRINT "Study this example and read the appropriate paragraph in your"  
5620 PRINT "manual."  
5630 PRINT  
5640 INPUT "press ENTER":$  
5650 GOSUB 21900  

244
£00r
PRINT, nuletell
£92
PRINT 6Z'
PRINT 0
OUT
have to EDIT the above line cat'at".
og
print the inappropriate command for editing line 10. then move the 's'
spacebar over to the appropriate place, then delete the two'
offending characters, then press ENTER" (use upper case)"
PRINT
INPUT ENTER the first command now IT%
PRINT
IF "EDIT" THEN PRINT "WRONG - you should type in EDIT 1) first":PRINT:GOTO 460
PRINT:": I =
PRINT: AS = INKEY%:
IF AS = "#" THEN GOTO 550
IF AS = "f" THEN AS = "#"
IF AS = CHRS(27) AND I = 3 THEN I = 1 : IF PRINT MID$ OF AS,1,1=gosd 5500
IF AS = "C" AND I = 3 THEN PRINT MID$ OF AS,R AND I = 12: GOTO 5500
IF AS = "C" AND I = 12 THEN PRINT MID$ OF AS,12,11=15: GOTO 5500
IF AS = "C" AND I = 15 THEN PRINT:PRINT: "WRONG - do not hit ENTER until you are done": PRINT:PRINT: "TRY AGAIN": PRINT:GOTO 640
IF AS = "C" AND I = 15 THEN GOTO 5500
GOTO 5500:
PRINT: "PRINT NEXT 2":
PRINT: "PRINT: The line in the computer's memory would now look like this": PRINT: "FOR I = 1077 : PRINT +
NEXT 2":
PRINT:
PRINT: "PRINT:GREAT ... if you want to do it again, press 'Y' else press 'N'":
INPUT ENTER your choice now 'Y' or 'N'":
IF "Y" THEN GOTO 8240
GOSUB 21000:
PRINT* nDelete*
PRINT:
PRINT: "For the example, we protected you from mistakes by ignoring":
PRINT: "some commands, and telling you what you did wrong for others."
PRINT: "if you really are editing a line, be sure you press the right'":
PRINT: "buttons, because you won't get warning messages' (however.)",
PRINT: "you seldom run what you have done, the editor is very for-"
PRINT: "it usually leaves you something, even when you make a'":
GOTO "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 0, then that is how many characters will be"
6780 PRINT"deleted."
6790 PRINT
6800 INPUT"press ENTER*"$*
6810 GOSUB 21000
6820 PRINT #6
6830 PRINT
6840 PRINT"If we wanted to delete 's*' all at once, we could position the"
6850 PRINT"cursor to just before the 's' and type in 20 and both "
6860 PRINT"characters would appear like this: 's**'
6870 PRINT
6880 PRINT"Now the exclamation marks, in this case, are around both"  
6890 PRINT"characters. When you get through here, practice with some"  
6900 PRINT"lines you have arbitrarily made up. You will see that this"  
6910 PRINT"command can be very handy."
6920 PRINT
6930 INPUT"press ENTER*"$*
6940 GOSUB 40000
6950 IF $s = "$* THEN GOTO 2000
6960 RETURN
6970 GOSUB 21000
6980 PRINT#
6990 PRINT "Xtend line!"  
7000 PRINT  
7010 PRINT "This is one of the easiest, and most useful of the commands"
7020 PRINT "It allows you to start at the end of a line, just as if you"
7030 PRINT "never pressed ENTER."
7040 PRINT
7050 PRINT "First, you go to EDIT mode, then you press x. When you do, "
7060 PRINT "you will see the whole line displayed, and you can add anything"  
7070 PRINT "on to the end of it. It is now. First, type the command to "
7080 PRINT "get into EDIT mode for line 10, then press x, then type in:"
7090 PRINT "anything you want (most micros allow a maximum of 249)"
7100 PRINT"characters!). Then press ENTER. Do it now."
7110 PRINT  
7120 GOSUB "FOR X = 107" :PRINT Y :NEXT Z"  
7130 INPUT"ENTER the first command!"$*
7140 IF $s = "$* THEN PRINT:PRINT "WRONG - you should type in EDIT 10 first"(PRINT:GOTO 715)
7150 PRINT
7160 PRINT"10 "
7170 A$ = INKEY$  
7180 IF A$ = "c" THEN A$ = "r"
7190 IF A$ = "r" THEN GOTO 7190
7200 PRINT #6
**** Listing of Program 'LESSON6A' ****

7210 LINE INPUT T$
7220 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 G$ + T$
7280 PRINT
7290 PRINT "Neat huh? You will find yourself using this command the most."
7300 PRINT
7310 INPUT "Press ENTER": T$
7320 GOSUB 40000
7340 IF T$ = "B" THEN GOTO 700
7350 RETURN
7360 GOSUB 21000
7380 PRINT "n(Change) & (Insert)"
7390 PRINT
8020 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 G$ = "FOR X = to77 :PRINT Y :NEXT Z"
8080 PRINT G$
8090 PRINT
8100 PRINT "Above is our line (with the 's*' characters missing - we"
8110 PRINT "deleted them in the n(Delete) section). Let's say we want"
8120 PRINT "to change 'NEXT Z' to 'NEXT Y' 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 (i.e. we want to insert a I just before the to77)"
8150 PRINT
8160 INPUT "Press ENTER to start our EDITing": T$
8170 GOSUB 21000
8180 PRINT "n(Change) & (Insert)"
8190 PRINT
8200 PRINT G$
8210 PRINT
8220 PRINT "First we'll change 'PRINT Y' to 'PRINT I'"
8230 PRINT
8240 PRINT "To use the n(Change) command, get into the EDIT mode and"
2250 PRINT"position the cursor to JUST BEFORE the character to be changed"
2260 PRINT"then press 'C' and press the new character, then press ENTER."
2270 PRINT"Try it now. Get to EDITor, SPACE over to just before the Y"
2275 PRINT"and type in 'C'. then type in an X we want to swap Y with it:
2280 PRINT"and finally, press ENTER (use capitals)"
2290 PRINT
2300 PRINT"ENTER the first command at the bottom of the next line"
2310 PRINT"10 *D1%
2320 PRINT"
2330 LINE INPUT T$
2340 IF T$ = "EDIT 10" THEN PRINT"WRONG - you have to type EDIT 10 first":PRINT: GOTO 2350
2350 PRINT
2360 I = 0
2370 PRINT"10 ";
2380 AS = INKEY$
2390 IF AS = "" THEN GOTO 3400
2400 IF AS = "C" THEN AS = "E"
2410 IF AS = CHR$(32) AND I < 20 THEN I = I + 1: PRINT MID$(T$,.I,.I):GOTO 3400
2420 IF AS = "E" AND I < 20 THEN PRINT:PRINT"WRONG - you must press 'C' just before the Y in PRINT"
2430 IF AS = "C" AND I = 20 THEN GOTO 3400
2440 GOTO 2350
2450 PRINT"
2460 AS = INKEY$
2470 IF AS = "" THEN GOTO 3400
2480 IF AS = "E" THEN AS = "C"
2490 IF AS = CHR$(13) THEN GOTO 3460
2500 PRINT"\n\n10 NEXT I"
2510 GOTO 300
2520 PRINT
2530 PRINT"How's that for class? Remember, we protected you from mistakes."
2540 PRINT"The real editor will do whatever you tell it, even if it is"
2550 PRINT"wrong. But you knew that, didn't you?"
2560 PRINT
2570 INPUT": PRINT"Enter a 'Y' if you want to do this again, else ENTER an 'N'":
2580 IF T$ = "Y" THEN GOTO 3170
2590 IF T$ = "N" THEN GOTO 3550
2600 GOSUB 2300
2610 PRINT"
\n\nchange) & Insert)"
2620 PRINT
2630 PRINT"As with the other commands, the 'n' in n(\n\n2640 PRINT"how many characters are affected by the command. If you want"
2650 PRINT"to change 10 characters, then you would type '10C' in the EDIT"
2660 PRINT"mode, and you would then HAVE to change the next 10 characters."
2670 PRINT
*** Listing of Program 'LESSON6A' *** 07/10/83 - 04:29:39

3660 PRINT "What would you type if you were in the EDIT mode and your "
3670 PRINT "cursor was just before a block of 4 characters that you wanted"  
3700 PRINT "to change to 'XXX'"  
3710 PRINT  
3720 INPUT "ENTER your answer now":T$  
3730 PRINT  
3740 IF T$ <> "4CXXXX" THEN PRINT "WRONG - you should have typed 4CXXXX":PRINT:PRINT "TRY AGAIN":INPUT "press ENTER":T$:S020 8570  
3750 PRINT "GREAT! Now you have the idea!"  
3760 PRINT  
3770 INPUT "press ENTER":T$  
3780 GOSUB 21000  
3790 PRINT  
3800 PRINT  
8810 PRINT "*10 FOR X = to77 :PRINT X :NEXT I*"  
8820 PRINT  
8830 PRINT "We would change the 'Z' to an 'I' in the same way, but, to"  
8840 PRINT "save time, we'll invoke some magic, and change it now so we can"  
8850 PRINT "get to the INSERT command! READY? ???????????????"  
8860 PRINT "<< POOF >>. There, it's changed now. Look below."  
8870 PRINT  
8880 PRINT "*10 FOR X = to77 :PRINT X :NEXT I*"  
8890 PRINT  
8900 PRINT "Now would you like to have THAT editor at your command?"  
8910 PRINT "We'll now get to the (INSERT) command. Remember, we want to"  
8920 PRINT "INSERT a 'I' just before the 'to77'."  
8930 PRINT  
8940 PRINT "press ENTER for the insert example":T$  
8950 GOSUB 21000  
8960 PRINT  
8970 PRINT "*10 FOR X = to77 :PRINT X :NEXT I*"  
8980 PRINT  
8990 PRINT  
9000 PRINT "For I = to77 :PRINT X :NEXT X"  
9010 PRINT "O"  
9020 PRINT  
9030 PRINT "To use the (INSERT) command, you first get into the EDIT mode"  
9040 PRINT "and then place the cursor to just before the character you want"  
9050 PRINT "to insert the data in front of."  
9060 PRINT  
9070 PRINT "In this case, we get into the EDIT mode, then"  
9080 PRINT "SPACE over to just before the 'to77' and then we type an 'I'"  
9090 PRINT "(for INSERT). After the 'I' command we want to put in a 'I'."  
9100 PRINT "but we COULD type in as many characters we want ...."  
9110 PRINT "Until we press ENTER. At that time, all our changes are made"  
9120 PRINT "and we are returned to the IMMEDIATE mode."  
9130 PRINT  
9140 INPUT "press ENTER to start the example":T$
*** Listing of Program 'LESSON6A' ***

9170 50SUB 21000
9180 PRINT"Reminder, first type EDIT 10, then space over to just before"
9190 PRINT"the 'to7'. then type '"'I'. then type a 1. then type ENTER."
9200 PRINT
9210 PRINT"-10"$10
9220 PRINT
9230 INPUT"ENTER the first command"$10
9240 IF I$ = "EDIT" THEN PRINT "WRONG - you must type EDIT 10 first":PRINT:S:GOTO 9230
9250 PRINT
9260 PRINT"10"$10
9270 I = 0
9280 IF A$ = "INPUT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PRINT$PR
7650 GOSUB 40000
7660 IF T$ = "B" THEN GOTO 8000
7670 RETURN
10000 GOSUB 21000
10100 PRINT "You have finished the lesson and you can now take the test.
10101 PRINT "If you wish to review parts of the lesson, ENTER an "R"
10102 PRINT "else, if you want to continue to the test ENTER a "C"
10103 PRINT
10140 INPUT "ENTER your choice now :R or C":T$
10150 IF T$ = "R" THEN GOTO 8000
10160 IF T$ = "C" THEN GOTO 10000
10170 GOTO 47000
20980 REM **
20990 REM *** This subroutine clears the screen on any terminal
21000 REM **
21100 FOR I = 1 TO 24
21200 PRINT
21210 NEXT I
21300 PRINT
21400 GOSUB 21000
21500 PRINT "Lesson 6"
30100 PRINT
30110 PRINT "This is the second part of a two part lesson"
40200 PRINT "It is divided into the following sections."
50200 PRINT
50300 PRINT ": 1) Introduction 4) nDelete"
50400 PRINT ": 2) Starting (EDIT/exit) 5) xItend line"
50500 PRINT ": 3) nSPACEBAR, nSearch, n(Change), nInsert"
50600 PRINT ": 6) TEST"
50700 PRINT
50800 RETURN
10100 GOSUB 21000
10110 PRINT "Which do you wish to do?"
10120 PRINT
10130 PRINT "* A) Continue on"
10140 PRINT "* B) Review this section again"
10150 PRINT
10160 INPUT "Press the letter opposite the correct answer and press ENTER":T$
10170 IF T$ = "A" OR T$ = "B" THEN RETURN
10180 GOTO 40000
40000 PRINT
40100 PRINT
40200 PRINT "You are now testing. Please wait a moment."
40300 RUN "TEST.X"
1000 REM **
1010 REM ** LESSON: TESTb
1020 REM ** AUTHOR: CAPT DAN CREATAN
1030 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM ** N$: 1: = NAMES ARRAY, USED TO READ IN SEQ-
1070 REM ** UENTIAL NAMES, AND TO WRITE OUT
1080 REM ** UPDATE NAMES.
1090 REM ** S(I) = SCORES ARRAY - USED TO READ AND
1100 REM ** WRITE SCORES
1110 REM ** Z(I) = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM ** CORRECT ANSWERS. IF AN ARRAY
1130 REM ** ELEMENT EQUALS 1, THE ANSWER WAS
1140 REM ** CORRECT
1150 REM **
1160 CLEAR 2000
1170 GOSUB #110
1180 DIM N$(1000)
1190 DIM Q(100)
1200 DIM B(1000)
1210 PRINT "FINAL TEST (lesson a:"
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:"
1310 GOSUB #110
1320 PRINT "What is wrong with the following statement?"
1330 PRINT
1340 GOSUB #430
1350 PRINT
1360 PRINT "Multi Statement lines are not allowed"
1370 PRINT "The assignment of values between A$ and B$ are not valid."
1380 PRINT "The strings were not initialized"
1390 PRINT "Nothing"
1400 PRINT
1410 GOSUB #431
1420 PRINT
1430 IF "$ = 'C'" THEN GOTO 142)
1440 PRINT "Wrong - the correct answer is D"
1450 PRINT "See part 1. String Functions"

262
***** Listing of Program "TEST6" *****

1460 GOTO 1470
1470 PRINT "CORRECT"
1480 Q(1) = 1
1490 PRINT
1500 INPUT "press ENTER"; T$ 
1510 GOSUB 4110
1520 GOSUB 4220
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$ = "M" THEN GOTO 1640
1600 PRINT "WRONG - the correct answer is M"
1610 PRINT "If A$ = H$ 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 "FOR X = 1 TO 20"
1720 PRINT "A$(X) = "CHR$(34) + "CHR$(34)"
1730 PRINT "NEXT X"
1740 PRINT
1750 PRINT "The array is not dimensioned properly."
1760 PRINT "You cannot address a single dimensioned array with a loop."
1770 PRINT "The "$ should not be enclosed in quotes." 
1780 PRINT "Nothing" 
1790 PRINT
1800 GOSUB 4230
1810 PRINT
1820 IF T$ = "A" THEN GOTO 1880
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?"
**** Listing of Program 'TEST6' ****

1920 PRINT
1930 GOSUB 4270
1940 PRINT "B$ = A$ + B$"
1950 PRINT
1960 INPUT "ENTER your answer EXACTLY as it would appear": T$
1970 PRINT
1980 IF T$ = "Hi!": THEN GOTO 2030
1990 PRINT "WRONG - the correct answer is Hi!"
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"
2140 Q(4) = 1
2150 PRINT
2260 INPUT "press ENTER": T$
2270 GOSUB 4110
2280 PRINT "What is the output of the following program?"
2290 PRINT
2310 PRINT "A$ = 'CHR$(34)"" SOMEWHERE"CHR$(34)'; AS = MID$(AS,1,4); PRINT AS"
2330 PRINT
2340 INPUT "ENTER your answer EXACTLY as it would appear": T$
2350 PRINT
2360 IF T$ = "SOME": THEN GOTO 2180
2370 PRINT "WRONG - the correct answer is SOME"
2380 PRINT "See your BASIC manual."
2390 GOTO 2200
2400 PRINT "CORRECT"
2410 Q(5) = 1
2420 PRINT
2510 INPUT "press ENTER": T$
2520 GOSUB 4110
2530 PRINT "What is the command you would enter to edit line number 50 of?"
2540 PRINT a program"
2550 PRINT
2620 INPUT "ENTER your answer EXACTLY as it would appear": T$
2630 PRINT
2640 IF T$ = "edit 50": THEN GOTO 2230
2650 IF T$ = "EDIT 50": THEN GOTO 2230
2660 PRINT "WRONG - the correct answer is EDIT 50"
2670 PRINT "See part 2, EDIT"
2680 GOTO 2250
2690 PRINT "CORRECT"
2740 Q(6) = 1
2750 PRINT
2820 INPUT "press ENTER": T$
2870 GOSUB 4110
Listing of Program 'TEST6'

07/10/83 - 04:48:04

2380 PRINT "Assume you are in the EDIT mode. You wish to place the "
2390 PRINT "cursor over the second occurrence of the letter R in your line."  
2400 PRINT
2410 PRINT "What is the command you would use?"
2420 PRINT
2430 PRINT "A 2DR"  
2440 PRINT "B 2DR"  
2450 PRINT "C 2DR"  
2460 PRINT "D 2DR"  
2470 PRINT "E  spacebar R"  
2480 PRINT
2490 GOSUB 4:180  
2500 PRINT
2510 IF T$ = "A" THEN GOTO 2550  
2520 PRINT "WRONG - the correct answer is A"  
2530 PRINT "See part 2, INSERT!"  
2540 GOTO 2570  
2550 PRINT "CORRECT"  
2560 Q(I) = 1  
2570 PRINT
2580 INPUT "Press ENTER":T$  
2590 GOSUB 4:110
2600 PRINT "Assume you 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 "Press 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, INSERT!"  
2700 GOTO 2730  
2710 PRINT "CORRECT"  
2720 Q(I) = 1  
2730 PRINT
2740 INPUT "Press ENTER":T$  
2750 GOSUB 4:110  
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

255
***** Listing of Program 'TEST6' *****

2840 IF T$ = "1" OR T$ = "z" THEN GOTO 2880
2850 PRINT "Wrong - the correct answer is 1"
2860 PRINT "See part 2, xtend)"
2870 GOTO 2900
2880 PRINT "Correct"
2890 Q:I: = 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 2(10) = 1
3020 GOSUB 4130
3030 PRINT
3040 IF T$ = "2" THEN PRINT "Don't blame me ... I'm only following orders"
3050 IF T$ = "3" THEN PRINT "That won't help the National Debt very much"
3060 IF T$ = "4" THEN PRINT "Don't try to butter me up, I know you're into masochism"
3070 IF T$ = "5" THEN PRINT "Gobvious: we have failed to communicate. I'm reporting you to the FBI
3080 PRINT "for tax evasion and mail fraud."
3090 PRINT "Try again"
3100 PRINT "Correct:
3110 PRINT "Obviously, you get automatic credit for the last question."
3120 PRINT
3130 PRINT "Congratulations, you are one of the few who selected an answer"
3140 PRINT "that wasn't listed. Were you ever an extra for the MUPPET SHOW?"
3150 PRINT
3160 PRINT
3170 INPUT "press ENTER":T$
3180 GOSUB 4110
3190 FOR I = 1 TO 10
3200 Y = Y+1
3210 NEXT I
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"
***** Listing of Program 'TEST7' *****

2200 GOSUB 2220
2220 IF X = 'n' THEN GOTO 2250
2230 PRINT "YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS";
2240 PRINT;
2250 PRINT "YOU SHOULD RETAKE LESSON 5";
2260 PRINT;
2270 PRINT "You will be returned to the Menu.
2280 PRINT
2290 GOTO 4250
2300 PRINT
2310 PRINT "Do you want your score recorded on a permanent file?";
2320 PRINT;
2330 PRINT "YES";
2340 PRINT "NO";
2350 PRINT
2360 INPUT "Which?"; S
2370 IF S = "B" THEN GOTO 3700
2380 GOSUB 4110
2390 PRINT "To record your score, we must have a file and a name or your score will not
2400 PRINT "be recorded. Therefore, we need your score, your name, and a file name.
2410 PRINT "If you do not have a file name, we will use your name as a file name.
2420 PRINT "Please contact your test director for an identifying word that will
2430 PRINT "make your name unique. Then enter that word below."
2440 PRINT;
2450 PRINT $T: IF S = "B" THEN GOTO 3700
2460 PRINT "If you have already entered a score previously, be sure to
2470 PRINT "enter the same name you used before. Use all capital letters.
2480 PRINT
2490 PRINT
2500 INPUT "ENTER your last name or name now?"; S
2510 OPEN 11,..."SCORES".
2520 I = 0
2530 IF EOF(11) THEN GOTO 3520
2540 I = I + 1
2550 INPUT "\$l"; S
2560 INPUT \$"=11'
2570 IF \$"=I" THEN GOTO 3300
2580 GOTO 3300
2590 CLOSE
2600 I = I + 1
2610 \$"=I" = "\$"=I"
2620 OPEN 11,..."SCORES".
2630 FOR I = 1 TO X
2640 PRINT \$"=I";
2650 PRINT \$"=I"
2660 NEXT I
2670 END
**** Listing of Program "TESTs" ****

100 10:30 4110
1770 PRINT "You are FINISHED WITH THIS COURSE.
1771 PRINT
1772 PRINT "It was GREAT having you as a student - THANKS!"
1773 PRINT "When you press ENTER you will be sent to the Menu from the
1774 PRINT "where you can review other lessons or quit."
1775 PRINT
1776 PRINT "For homework . . . you may wish to change your inventory program",
1777 PRINT "so that it will handle string data. That way you can include",
1778 PRINT "the names of your furniture in your file. However, we leave",
1779 PRINT "that up to you. Hasta Luego!"
1780 PRINT
1781 GOTO 2500
1782 S = 1
1783 IF EOF(1) THEN CLOSE: GOTO 3500
1784 INPUT#, N$(1, 40):
1785 GOTO 3500
1786 OPEN"D:1."SCOREs"
1787 FOR I = 1 TO T
1788 PRINT#u, N$(1, 40)
1789 PRINT#u, S$(1, 40)
1790 NEXT I
1791 PRINT
1792 GOTO 2500
1793 IF #10 THEN RETURN
1794 PRINT "You NEED IMPROVEMENT IN THE FOLLOWING AREAS!"
1795 PRINT
1796 IF S = 0 THEN PRINT "part 1. STRING ASSIGNMENT, STRING FUNCTIONS"
1797 IF L = 0 THEN PRINT "part 1. STRING ASSIGNMENT, STRING FUNCTIONS"
1798 IF D = 0 THEN PRINT "part 1. STRING ARRAYS"
1799 IF 0 = 0 THEN PRINT "part 1. CONCATENATION"
1800 IF 0 = 0 THEN PRINT "part 1. STRING FUNCTIONS"
1801 IF 0 = 0 THEN PRINT "part 1. EDIT"
1802 IF 0 = 0 THEN PRINT "part 1. SEARCH"
1803 IF 0 = 0 THEN PRINT "part 1. INSERT"
1804 IF 0 = 0 THEN PRINT "part 1. DELETE"
1805 PRINT
1806 INPUT "Press ENTER for"
1807 PRINT 4111
1808 RETURN
1809 END **
1810 DEF ** "This subroutine clears the screen"
1811 DEF **
1812 FOR I = 1 TO 24
1813 PRINT
1814
*Listine of FraCrat

4130 LIST
4140 RETURN
4150 REM **
4150 REM ** this subroutine is for the response section
4150 REM
4160 INPUT "ENTER the letter opposite the correct answer":R$
4170 RETURN
4180 REM **
4190 REM ** subr for string assignment/functions
4200 REM **
4210 PRINT 1: AS = "CHR$(14)" + "CHR$(14)" : BS = AS + BS = LEFT$(BS, 1): PRINT BS$
4220 RETURN
4230 PRINT
4240 INPUT 'press ENTER to return to MENU':R$
4250 RUN 'MENU.R'
4260 CLOSE
4270 OPEN 1, "SCORES"
4280 IF EOF THEN 9999
4290 INPUT A,N
4300 PRINT A
4310 B = INT (.4 + A)
APPENDIX C

OUTPUT OF THE CAI PROGRAM LIBRARY
# Table of Contents

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>261</td>
</tr>
<tr>
<td>Lesson One</td>
<td>262</td>
</tr>
<tr>
<td>Lesson Two</td>
<td>305</td>
</tr>
<tr>
<td>Lesson Three</td>
<td>348</td>
</tr>
<tr>
<td>Lesson Four</td>
<td>381</td>
</tr>
<tr>
<td>Lesson Five</td>
<td>414</td>
</tr>
<tr>
<td>Lesson Six</td>
<td>445</td>
</tr>
</tbody>
</table>
TRSDOS Ready
BASIC
BASIC v1.00.30 for TRSDOS Version a
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 feedback or get information about this program, please contact me at GRIFFIS 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
2. TEST ONE
3. LESSON TWO
4. TEST TWO
5. LESSON THREE
6. TEST THREE
7. LESSON FOUR
8. TEST FOUR
9. LESSON FIVE
10. TEST FIVE
11. LESSON SIX
12. TEST SIX

WHICH NUMBER DO YOU WANT?
LESSON: BASIC 1A  
VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON: About One Hour

AUTHOR: Capt Danny J. Craigan  
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  a) Print Statement
3) Software  b) End & Stop Statement
4) General Information  c) 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

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
Ink 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 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.
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
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)

269
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:

```plaintext
PRINT (19+30+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?

What 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 65535 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.

(lock 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.
pres 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

correct the letter beside the correct answer and then press ENTER? Y
**Listing of Program 'LES0N1.TXT'**

07/11/83 - 00:04:30

WRONG - 250000 is too big. Remember, you can only go to $5529

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

1) 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:
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 3 spaces between terms.

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

Results in:

```
A    B    B    B
```

Now is the time to skip
**** Listing of Program 'LESSON1/T1' ****

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
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 = 3/5
30 STOP
40 C = 1+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 STOPped it, then type in CONT (CONTinue) and press ENTER. For example:
### Listing of Program 'LESSTXT' ###

```
10 X = 12+3
20 Y = J*2
30 STOP
40 PRINT X+Y
50"

Results in:

BREAK IN 20

hit ENTER for the rest

BREAK IN 20

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

CONT
20

Notice how the last line (which was PRINT X+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 can 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 95/10=9.45

Which would result in:

14.5

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 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
***** Listing of Program 'LESSON1/TXT' *****

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: IF a.

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

CORRECT! HUNGO - 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.
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  4) String Variables
2) Variables (general) 5) Using Arithmetic
3) Numeric Variables  6) Comparing Variables
7) TEST

A. I'm taking this part in its entirety.
B. I wish to review selected areas (or take the test).
Listing of Program 'LESSON1/TXT' 

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, trigonometric functions, and logarithms 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 SQRT 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 SQRT(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
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

2011 time: USE ONLY CAPITAL LETTERS IN YOUR ANSWERS!
Are the library functions stored permanently in memory?
A: yes
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 handling, if for no other reason than to communicate your program to someone else. Numbers, such as 10. 22. 39. 1, etc., are considered constants. Can you guess why? It's 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 these 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, save word processors, and develop computer assisted instruction programs. If these tasks are done in BASIC, then they are done using STRINGS.

press ENTER
**Listing of Program "LESSON1.TXT"**

Which do you want to do?

A Continue on
B Review this lesson again

dress the letter opposite your choice and dress 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 x = 3+2
20 x = 4
```

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

dress 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 SEENDING letter OR number to a variable to distinguish it from another. A1, A2, Y1, V1, and FF are legal variables. I, II, or III are not legal. Can you see why? First, 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 I = 4
20 J = I
30 I = I+1

The variable I is assigned the value of 4 or 5 or 11.

All variables are assigned the value of 0 when you first start a Microsoft BASIC program. However, some languages assign unknown 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 is a very important 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

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 \textit{LET} to assign a value to a variable.

10 \textit{LET} \( x = 4 \)

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

Press \textit{ENTER}.

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

A 10 \textit{LET} \( x = 2 \)
B 20 \( x = 2 \)
C 10 \( x = 2 \)
D 10 \( x = \times \)

Press the letter opposite the correct answer and press \textit{ENTER}.

**CORRECT** - Good job!

Press \textit{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 100 \( C = A + B \)
D 10 \( C + C = A + B \)

291
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?
4 Continue on
B Review this lesson again

press the letter opposite your choice and press ENTER

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 taking 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:

$ = "the author is he"
press ENTER

String Variables cont.

$ = "the author is he"

Here, the variable label $ is identified as a STRING variable 
by the addition of a dollar sign. Further, the CHARACTER data 
"the author is he" is assigned to the STRING.
**** Listing of Program 'LESSON 71:TA' ****

C1: "$14 West Cottage Street"

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

Examples of illegal STRING labels would be $, &&, $CC$, $CC$, $CC$

dress ENTER

Which of the following are correct STRINGS

A $A$ = $1$
B $A$ = "$12"
C $A$ = "North State Street"
D $A$ = "North State Street"

dress the letter opposite the correct answer and dress ENTER B

CORRECT - are you sure you are only a student

dress ENTER

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

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

The is 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
mathematical 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-2  
*       | multiplication| A*B  
/       | division| A/B   
^       | exponentiation| A^2  : A squared;

Parentheses ( ) are also used, just as in algebra.

press ENTER?

Using Arithmetic (cont)

Note that a * always must be used for multiplication.
If you try to use an x or a, the computer
would think you were trying to put in another variable label.
It would give you an error message. Also, you cannot use
terms like 1/2 to mean 1/2. If you do, you will get an error.

194
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:

\[ t = (10-2.5-2*2) \]

In this statement, the computer will scan the line and do all terms within parentheses 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 listed below:

<table>
<thead>
<tr>
<th>Category</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>HIGHEST</td>
</tr>
<tr>
<td>* or /</td>
<td>Next HIGHEST</td>
</tr>
<tr>
<td>+ or -</td>
<td>LOWEST</td>
</tr>
</tbody>
</table>

press ENTER for more
Using Arithmetic (cont)

\[ x = 1 + \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \]

In the first scan, the computer would do the terms within the parenthesis. It would first do exponentiation \( \frac{1}{2} \), \( \frac{1}{2} \), 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

Parentheses 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 parentheses. For example:

\[ S = 12 \cdot (a+b) \]

is much different than:

\[ S = 12 \cdot a + b \]

Do you see why? The first value assigned to \( S \) is 12, the second value assigned is \( a \cdot b \). Sub. the example carefully. Press ENTER.

Another: the following statements will assign the value of \( a \) to the variable \( x \).
A. \( M = 1 \times 2(2+3)-2 \)
B. \( M = 5 \times 2(3+3)-10 \)
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. 
\[ M = 20 / (2(3+3)-1) \]
\[ M = 20 / (2 \times 6 - 1) \]
\[ M = 20 / 11 \]

Press ENTER to continue.

Which do you want to do?

4. Continue on
5. 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.
### Listing of Program "LESSON1.TXT" ###

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>equal</td>
<td>A=2</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>A&lt;B</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>A&gt;B</td>
</tr>
<tr>
<td>!=</td>
<td>not equal to</td>
<td>A!=B</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal</td>
<td>A&lt;=B</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal</td>
<td>A&gt;=B</td>
</tr>
</tbody>
</table>

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.

```plaintext
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.
***** Listing of Program 'LESSON1/TXT' *****

1) Library Functions   4) String Variables
2) Variables (general) 5) Using Arithmetic
3) Numeric Variables   6) Comparing Variables

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.

---

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

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?

---

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 ques-
tions). 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!

Please enter to continue.

Is a computer program called Software?
A  Yes
B  No

Please enter the correct answer and press ENTER. A

CORRECT

Please enter to continue.

Which of the following is an example of a FUNCTION
A  LIST
B  NEW
C  SUB
D  ADD

Please enter the correct answer and press ENTER. C

CORRECT

Please enter to continue.

Which statement would print the word TEST
A  PRINT TEST
B  PRINT "TEST"
C  OUTPUT "TEST"
D  PRINT 'TEST'

Please enter to continue.
***** Listing of Program 'LESS0147A' ***** 07/11/83 - 15:45:22

Press the letter opposite the correct answer and press ENTER b

CORRECT
Press ENTER

What would the following program's output be?

10 X = SCR(A)
20 PRINT E - X
RUN

Type in your answer and press ENTER B

CORRECT
Press ENTER

Save 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?

1 TRUE
2 FALSE

Press the letter opposite the correct answer and press ENTER b

CORRECT
Which of the following statements is invalid?

A 25 = x
B PRINT SGR(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 T for TRUE or t for 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=34
20 T=E0
30 T=10
40 L=5

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 to 40

Press the letter opposite the correct answer and press ENTER

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 cut 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 proctor 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 have me? DAN CREGAN

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 01.06.90 for TRSDOS Version 3
Copyright © 1983 By Microsoft. Licensed to Tandy Corporation.
All rights reserved.

Ready
run "lesson2"

LESSON: BASIC I
VERSION: 1 AUGUST 93

TIME REQUIRED TO COMPLETE LESSON: About One Hour

AUTHOR: Capt. Danny J. Creegan
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

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 J, L, AND 22 IS"
20 PRINT J + L + 22
30 PRINT
40 PRINT
50 PRINT "THE SUM OF J, L, 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 2, 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 ORGMEMCO (or CPM)

.filename/.extension (filename/.extension)

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

press ENTER

Filenames (cont)

TRS-80 ORGMEMCO
(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 .extACTER. 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
SWIMFIN.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  44444444.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  44444444.FIL!
press ENTER*

Is the extension necessary for a filename to be legal?
***** Listing of Program 'LESSON2/TIT' *****

A YES
9 NO

Press the letter opposite the correct answer and press ENTER?

WRONG - the correct answer is 9 (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 'BOOM' (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 DISKEMCO hard
disk, you will probably not have any problem in this area. Ask
your SYSTEM OPERATOR for more information. If you are using
"20-M" small floppy disk, then go to the COMMAND mode by
typing in CMD-5 and then typing in DIR 1 or
DIR 2. Drive 1 is called 1 and drive 2 is called 2. 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 crams, 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?
**** Listing of Program 'LE56P216' ****

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 SAVING 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 PRITI?

A. SAVE PRITI
B. LOAD PRITI.BAS
C. SAVE MYPROG
D. SAVE PRITI

Press the letter opposite the correct answer and press ENTER.

WRONG - the correct answer is D (SAVE PRITI).

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. You would have to load the program from permanent storage to temporary memory. Which of the following would you type in:
**Listing of Program "LESSON27"**

```
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.
```

**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. REM - statements.

Press ENTER.

**Remarks cont:**

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

An example is:

```
```
***** Listing of Program 'LESSON4.TXT' *****

10 REM This is how a REMark statement is made
20 WHEN the computer sees REM it ignores all data that follows
30 press ENTER

Here is an example of REMarks in a program:

50 C = 1
60 REM the variable C in the next line is approx equal to Pi
70 PI = 3.14
80 PRINT C
90 REM

which would give:

3.14

Notice that the REM statement was not printed

press ENTER

100 PRINT "SWEETMRS is my girrl.
110 REM SWEETMRS is my girrl.
120 PRINT "Her real name is CANDY
130 REM "I love her"
140 REM

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

SWEETMRS is my girrl.
Her real name is CANDY
"I love her"

TRUE OR FALSE, whichever is correct

WARNING - the correct answer is TRUE - REM statements are not printed

110
**** Listing of Process 'LESSON1.TXT' ****

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 I = 10
20 PRINT I
RUN

Gives 10.

In this example, we assigned 10 to I. 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
***** Listing of Program "LESSON2.CAS" *****

Press ENTER

INPUT Statements (cont)

We INPUT "Press the correct letter, then press ENTER" twice.

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 "x", 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:

1) A = 2
2) INPUT "Enter a number between 1 and 9" PRINT
3) C = A + x
4) PRINT

Gives us:
Enter a number between 1 and 9
if we ENTER a 2 then

Press ENTER

INPUT Statements (cont)
INPUT ENTER a number between 1 and 7

we have learned 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 can 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:

1) PRINT "when you see a question mark, ENTER a 5"
2) INPUT
   FUN

Gives you:

when you see a question mark, ENTER a 5

press ENTER

INPUT Statements (cont.)

1) PRINT "when you see a question mark, ENTER a 5"
2) INPUT
   FUN

when you see a question mark, ENTER a 5
Notice now the INPUT statement brood (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)

1: INPUT ENTER a number between 1 and 9

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 gave 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 'SYSE' 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, put the right-hand variable on the end of the INPUT statement. You will be tested on this.

press ENTER
INPUT Statements (cont.)

You can 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 inserted for each variable.
For example:

10 INPUT ENTER three 3. numbers 'A,B,C
   A,B,C

ENTER three 3. numbers? 10
   22
   6

press ENTER

INPUT Statements (cont.)

10 INPUT ENTER three 3. numbers 'A,B,C
   A,B,C

ENTER three 3. numbers? 10
   22
   6

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 starred data input.

press ENTER

INPUT Statements (cont.)

10 INPUT ENTER three 3. numbers 'A,B,C
   A,B,C
**** Listing of Program 'LESSON1.TEX' ****

ENTER three (5) numbers: 10.12.5

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

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

cross 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.

cross ENTER!

What is the prompt that an INPUT statement ALWAYS gives?

A: A question mark
B: "no question marks"
C: Space
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

cross ENTER!

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

C: INPUT A B C
***** Listing of Program "LESSON.18" *****

A  C:ten.18
B  C:1.1.15 North Elia
C  C:1.1.16
D  C:2.NONE.9

ENTER the letter opposite the correct response and press ENTER.

WRONG - the correct answer is B: C:1.15 North Elia.

dress ENTER

which do you wish to do?

A  Continue on
B  Review this section again

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

GOING TO SECOND PART - PLEASE STANDBY

LESSON 18

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

1) READ. DATA and RESTORE
2) IF Statements
3) 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 code. 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.
An example of a READ statement that would read values into variables B1, A, and X$ is:

10 READ B1,A,X$

press ENTER

READ and DATA (cont)

10 READ B1,A,X$

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 X$ would hold 22, 1.5, 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 preceded 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,A1$
20 DATA 215.50,$"AIN'T she sweet"$

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 an
error at least it will ALWAYS give you an ERROR
message.
press ENTER

READ and DATA (cont.)

10 READ B1,A$
20 PRINT B1,A
30 PRINT "The average of these numbers is"$
40 PRINT (B1+A)/2
50 DATA 8.10,15
60 PRINT

Gives us:

8.10 15
The average of these numbers is 10

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

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
20 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
20 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, 5
```

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,I
20 DATA 25.2,15.55,64
30 PRINT I

A. Yes
B. No

Enter the letter opposite the correct answer:

WRONG - the program is valid

Press ENTER.

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

WRONG - the correct answer is 15. I is the third variable to be read, so the third data point is out 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.
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 A1.81
20 PRINT A1.81
30 RESTORE
40 READ C1.81
50 PRINT C1.81
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.

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 GOTO line 50 if the CONDITION that A = 5 is satisfied. That is, control will only be transferred to line 50 if A = 5.

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 GOTO line 20 when it is executed. There will be no choice made. Control will go to line 20.

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. Every time 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?
**Listing of Program "LESSON2.TXT"**

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 to quit!" IN
20 IF N = 0 THEN STOP

334
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 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 50. At that time, the check for N GREATER THAN 10 will be set and the message will be printed. No other condition will be set 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 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) < (2*10+6) THEN GOTO 19

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

30 IF (20*2-3) < (2*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 20

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)
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.

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 = 3 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 = 3 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 Z:IN
20 ON N GOTO 70,50,70
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 Z:IN
20 ON N GOTO 70,50,70
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 past the GOTO word. Control branches to the line number indicated by this \(N\)th 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.
LET N = 30.50.70
20 PRINT "YOU ENTERED A ONE (or an illegal number)"
40 STOP
50 PRINT "YOU ENTERED A TWO"
70 STOP
80 PRINT "YOU ENTERED A THREE"
90 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 50 and 40
B 50 and 80
C 70 and 80

Press the letter opposite the correct answer then press ENTER

W R O N G - the correct answer is A "50 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 29

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

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

4) I'm taking this part in its entirety.
2) I wish to review selected areas (or take the test).
1) I want to go to the first part.
0) I want to return to the Menu.

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

LESSON 29

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

1) READ, DATA and RESTORE 2) IF Statements
3) 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 more 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.

dress ENTER to continue.

Which of the following is a legal filename?

A. SCI111E.CMD (in CPM or Commodore)  
B. TRIUMPH/650 (in TRS-80)  
C. THEWAYOF.E21 (in CPM or Commodore)  
D. SCORE12/DAT (in TRS-80)

ENTER the letter opposite the correct answer.

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

dress 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 1.

press ENTER?

REM* 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
REM* 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":IN

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 40,60,20
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.
***** Listing of Program 'LESSON2/TIT' *****

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:

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

cart ENTER?
YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS
YOU SHOULD RETAKE LESSON 1.
you will be returned to the Menu.

dress ENTER to return to the MENU?

Break in RCD
Ready
SYSTEM RESET CODE
TRSDOS Ready

Lesson: BASIC 3
Version: 1 August 83

Time required to complete lesson: Less than one hour

Author: Capt. Danny J. Crestan
Air Force Institute of Technology

Objective: "Teach the student how to use LOOPS and ARRAYS.

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 & ILL
2. LOOPS: Intro
3. COUNTER variables
4. FOR NEXT Statements
5. Advanced FOR NEXT

4 I'm taking this part in its entirety.
5 I wish to review selected areas.
6 I want to go to the second part.
J. 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 phase 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 irreplaceable data or programs if used incorrectly. By now you should feel comfortable with some of the commands and you shouldn't take 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
**** Listing of Program 'LESSON:171' ****

Which of the following statements is legal?

A) KILL STUPID
B) KILL 'STUPID';
C) KILL "STUPID.ES"

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 accurately 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 7 columns of figures, the left column would be integers from 1 to 3, the middle column would be the square of the above in the adjacent first column.
and the third column would be the square of the figure in the "next" middle column. It would look like this:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>625</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>1296</td>
</tr>
<tr>
<td>7</td>
<td>49</td>
<td>2401</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>4096</td>
</tr>
</tbody>
</table>

Press ENTER.

If you were to write separate print statements for these calculations, you would have to write at least 8 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
```
**** Listing of Program "ENDER" ****

10 IF I = 10 THEN STOP
20 GOTO 10

The value of I after running this program is 11.
pres 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 every 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.
pres ENTER

To 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, KINS, and *** DUMMIES. You did it again: ***
pres ENTER
COUNTERs (cont)

If your program ever gets hung up, you may have to type
- CONTROL C + C or C-m or C-trapcol, 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 1).

CAUTION: If you are using this program on a multi-user system,
such as Crammed 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 n=0
20 x=0
30 y=1
40 IF X = 1 THEN I = I
50 IF X = 5 THEN STOP
60 PRINT I
70 GOTO 30
80 END

4 I is the COUNTER
B X is the COUNTER
C BOTH are COUNTERS, but Y will control the program

press the letter opposite the correct answer and press ENTER

CORRECT - that shows good understanding of the principle

press ENTER
**** Listing of Program "LESSON/EX" ****

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 FOR NEXT loop
10 REM
20 NEXT X
30 PRINT X
40 IF X = 3 GOTO 90
50 GOTO 20
70 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
Line 10 of the FOR NEXT loop initializes \( i \), the same way that lines 10 and 20 of the COUNTER loop do. The FOR NEXT loop sees line 10 as 'I am going to start a loop. \( i \) is the counter. I will begin with \( i = 1 \) and when \( i \) is GREATER than 3 I will GOTO the statement that is after the NEXT \( i \) statement'.

press ENTER

Line 20 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
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 \( x \) 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
Which of the following programs made the above list?

A: 10 FOR X=1 TO 3
   20 PRINT X*X
   30 NEXT X

B: 10 PRINT X*X
   20 PRINT X*X
   30 IF X=3 THEN GOTO 10
   40 GOTO 10

C: 10 FOR I=1 TO 4
   20 PRINT I*I
   30 NEXT I

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

dress 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.

dress ENTER
### Advanced FOR NEXT

10 FOR \( x = 10 \) to \( 1 \) STEP -2
20 PRINT \( x \)
30 NEXT \( x \)

Gives us:

10
9
8
7

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 \( y = 1 \) to \( 2 \)
20 FOR \( x = 1 \) 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)

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 first 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 1 statement caused control to GOTO line 10 again. The process was repeated, but X now was equal to 2.

press ENTER?

1 1
  2
  1
  2

10 FOR Y = 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 Y = 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)
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 I = 1 to 2
30 PRINT X*I
40 NEXT I
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 I = 1 TO 2
20 FOR J = 1 TO 3
30 PRINT "This is the last question"
40 NEXT J
50 NEXT I

A yes, it is valid
B no, it is not valid

ENTER the letter opposite the correct answer?

A - the program is valid
B - 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 3, 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 36
4 16 64
5 25 100
6 36 129
7 49 144
8 64 196

10 FOR I = 1 TO 3
20 PRINT I,I*I,(I*I)*I
30 NEXT I

It looks like child's play now, doesn't it
**** Listing of Program 'LESSON3/TXT' ****

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

4) I'm taking this part in its entirety.
3) I wish to review selected areas. I will take the test.
2) I want to go to the first part.
1) 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 databases 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.

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 categories (name, rank, training status) into three variables and use subscripts to provide a place for each entry in our database.

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Training Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Doe</td>
<td>Foreman</td>
<td>0</td>
</tr>
<tr>
<td>Jake Rodinski</td>
<td>Peon</td>
<td>3</td>
</tr>
<tr>
<td>Mark Muffin</td>
<td>Specialist</td>
<td>5</td>
</tr>
</tbody>
</table>

We could assign subscripted ARRAYS to the three main categories. 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)

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Training Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Doe</td>
<td>Foreman</td>
<td>9</td>
</tr>
<tr>
<td>Jake Rodinski</td>
<td>Peon</td>
<td>3</td>
</tr>
</tbody>
</table>
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.

```
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?

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Training Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Doe</td>
<td>Foreman</td>
<td>9</td>
</tr>
<tr>
<td>Jake Robinski</td>
<td>Peon</td>
<td>3</td>
</tr>
<tr>
<td>Mark Muffin</td>
<td>Specialist</td>
<td>5</td>
</tr>
</tbody>
</table>

This is what that program would output. Notice that we have
used only three variables, but we made the ARRays so that we
could hold nine bits of data.
As we continue through our lessons, we will discover some very
covrful uses for ARRays.
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 Microsot, will handle at least 3 dimensions. An example of a TWO dimension ARRAY would be A[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

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(2,2) will have all
numbers in the DATA statement. and the numbers will have
printed out.

Press ENTER?
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?

1. x=1, y=1
2. x=1, y=2
3. x=2, y=1
4. x=2, y=2

ENTER the letter opposite the correct answer

WRONG - the correct answer is C

press ENTER
If \( X = 1 \) and \( Y = 1 \), what value is in the pocket pointed to by \( \text{ARRAY A}(1,1) \)?

\[
\begin{array}{ccc}
A & 10 & 15 \\
B & 15 & 10 \\
C & 20 & 5 \\
D & 5 & 20 \\
\end{array}
\]

ENTER the letter opposite the correct answer.

WRONG - the correct answer is D

press ENTER?

\[
\begin{array}{ccc}
1. & 2. \\
\end{array}
\]

\[
\begin{array}{ccc}
1. & 5 & 10 \\
2. & 15 & 20 \\
\end{array}
\]

\( \text{ARRAY A}(i,j) \)

The FIRST subscript in an ARRAY ALWAYS indicates the ROWs of DATA, and the SECOND subscript indicates the COLUMNS of data. Therefore, \( \text{ARRAY A}(x,y) \) 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 1. you wish to do?

A) Continue on
B) Review this section again

case the letter opposite the correct answer and press ENTER.
Dimension & Clear

Whenever you use an ARRAY in BASIC, the processor has to make room for all the extra memory spaces that you will use. Up to a limit, the machine can handle unexpected ARRAYS, but after you designate more than 10 pockets or 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 5 dimensions to it (as N(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
90 RUN

1 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 & 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 space.
location. It must be able to find the letter once it stores it to 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 nearby 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 CLEAR 100
20 DIM G(50), A$(1000)
20 FOR X = 1 TO 100
40 READ A$(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 docket of A$(X)) and is going to use 50 docket of the numeric array G. (note that you always get docket 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(1)

A 0 because the clear statement is in the wrong place.
B 5 - the clear statement only affects STRINGS.
C Nothing because A(1): 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

CORRECT - Gooddddccc000000000000000D 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
the computer will not automatically reserve more space than needed. For example:

10 DIM A(10)

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(10)
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

TEST
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

374
Write out the first line of a FOR NEXT loop using \textit{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:

\textit{WRONG} - the correct answer is \texttt{50 FOR I = 1 TO 5 STEP 2}

See part 1, \texttt{FOR - NEXT}.

press ENTER

10 FOR \textit{X} = 1 TO 5 STEP 2
20 PRINT \textit{X};
30 \texttt{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: \texttt{1 3 5}

\texttt{CORRECT}

press ENTER

How many elements (or elements) can an array have without a DIM statement?

\begin{itemize}
  \item [A] 10 not counting the 0 element
  \item [B] 11 not counting the 0 element
  \item [C] 0 not counting the 0 element
  \item [D] 4 not counting the 0 element
\end{itemize}

ENTER the letter opposite the correct answer: \texttt{A}
***** Listing of Program 'LESSON3.FXT' *****

CORRECT
press ENTER?

How many sockets 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(0:1,0:1)

ENTER YES or NO? YES
CORRECT
press ENTER?

Which of the following is valid
***** Listing of Program 'LESSON 7/7' *****

A A$1
S (A$1)
C A$1:A$2
D (A1:1:2)

ENTER the letter opposite the correct answer? A

CORRECT

cress 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? TRUE

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.

cress 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. LOOPs
part 2. CLEAR statement

cress 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: RESET=30
TRSDOS Ready

BASIC
BASIC 31.00.00 for TRSDOS Version 6
Copyright © 1983 by Microsoft. Licensed to Tandy Corporation. All rights reserved.

TRSDOS Ready
BASIC
BASIC 31.00.00 for TRSDOS Version 6
Copyright © 1983 by Microsoft. Licensed to Tandy Corporation. All rights reserved.

Readv
RUN "LESSON4"

LESSON: BASIC 4
VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON: About 1.5 hours

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

OBJECTIVE: To teach the student how to make the computer communicate with standard peripheral devices.

cress 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 wish taking this part in its entirely.
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, S, C, or D and then press ENTER.

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 all the time. 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 taking your own program.

Remember, looking up the answers is not cheating, it's 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 Craystar, 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:

LPRINT "MONTHLY TRAINING REPORT"

1: A = 10
2: LPRINT A

Press ENTER.

LPRINT & LLIST

You can 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:

LPRINT TAB N, data
where k is a number between 1 and the length of your printer carriage. When used, the carriage will go over k columns before it starts to print. Some computers will not TAB past 30 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.

dress ENTER

LPRINT & LIST

LIST works just like LIST only it outputs to the printer

It is normally used from the IMEDIATE mode when you want to see your progress lines on paper. It is sometimes easier to find ‘bugs’ in your program if you can see it on a piece of paper.

LPRINT and LIST 100-400 are valid LIST commands. LIST 100-ALL lists lines 100-ALL to the printer.

dress ENTER

LPRINT and LU'LIST 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 B

CURRENT

dress ENTER
which of the following commands will LIST your entire program to the printer?

4. LIST
3. LIST ALL
2. LIST "ALL"
1. LIST "fn" (fn is your program name)

ENTER the letter opposite the correct answer 4

CORRECT

does ENTER

Which do you wish to do?

4. Continue on
3. Review this section again

press the letter opposite the correct answer and press ENTER 4

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 last 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 don't have to keep entering the same information everytime we run a program. We just have to tell our program to read the data from a disk. Our lives 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 ten 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.

- TRUE
- FALSE
- I DON'T KNOW

ENTER the letter opposite the correct answer - A

A) TRUE - You always have to create a BUFFER in memory.
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. Then it 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 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 3rd name you wrote without reading the first name you wrote?

A YES
B NO
C I wish to take the 5th

don't 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:
**** Listing of Program "LESSON4:TEST" ****

OPEN  PRINT#  EOR
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 a 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 "C:\\1\"TEST"
20 INPUT"Use your name and press ENTER!"IF$%
30 PRINT#1.7$%
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"?

1  TRUE
2  FALSE
3  "In between... somewhere in-between"

ENTER the letter opposite the correct answer: 2
## Listing of Program 'LESSON4.4' ##

**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.

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.

OPEN statement

10 OPEN "D:\L\TEST"
10 INPUT "Type in your name and press ENTER"$\n10 PRINT$"$\n40 CLOSE "$

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.
**** Listing of Program 'LESSON 4 : TAT' ****

message:  (also, don't OPEN a file that is already OPEN)

can you OPEN a file that is already OPEN?

1. YES
2. NO

ENTER the letter opposite the correct answer:  1

CORRECT

can you OPEN a file that is already OPEN?

10 OPEN 'D:\1\TEST'
20 INPUT Type in your name and press ENTER: 'T's
30 PRINT 'T's
40 CLOSE 1

line 10 OPEN the file in this manner:

OPEN is the keyword that tells 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 us to 3 buffers (more on some versions). For our pur-
poses, we will use buffer #1. The last bit of data is the
filename. Notice that the file mode and filename are in quotes

can you OPEN a file that is already OPEN?

10 OPEN 'D:\1\TEST'
20 INPUT Type in your name and press ENTER: 'T's
30 PRINT 'T's
40 CLOSE 1
***** Listing of Program 'LESSON4.TXT' *****

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
INPUTting data, the mode is 'I'. You can have up to 8 buffers
you must declare any buffers over 3 when working with the TRS-
30 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", "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.
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

1) OPEN "DOL1", "TEST";
2) INPUT "Give your name and press ENTER";
3) PRINT;
4) CLOSE ;

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 'LESSON' XL *****

A ONE   B TWO   C THREE   D FOUR

ENTER the letter opposite the correct answer: A

CORRECT

press ENTER

1. OPEN "C1", "TEST"
2. INPUT#1, "your name and press ENTER"
3. PRINT#, 1
4. 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 "C1", A$

press ENTER

10 A$ = "TEST"
20 OPEN "C1", 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 "C", "TEST"
## Listing of Program 'LESSON1.TXT'

```
A YES
9 NO

ENTER the letter opposite the correct answer? 9

CORRECT

press ENTER

10 OPEN "D:\"."TEST"
20 INPUT "Type in your name and press ENTER":T$  
20 PRINT 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 "D:\"."TEST"
20 INPUT "Type in your name and press ENTER":T$  
20 PRINT 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 and manipulated as one record or block at a time. You cannot just write a single record onto the front of the file that's on the disk.
```
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"S:\.J."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 "D"
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.

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.
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 to 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 odd).
***** Listing of Program 'LESSN4/T1T' *****

10 OPEN"O",J."QUESTION"
20 INPUT A$
30 PRINT#A,$
40 CLOSE
50 OPEN"O",J."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 A5 was CLOSED by line 40, so buffer 5 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 0

1: OPEN "O",J."TEST"
20 INPUT"Type in your name and press ENTER"TI$
30 PRINT"I.$
40 CLOSE 1

If line 30 we PRINT to the file buffer (that we OPENed in line 10) the variable I$. Notice that the number to the right of the PRINT$ corresponds to the buffer number. If the buffer was number 3, then, after we OPENed the file, we would PRINT$3.I$ 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 PRINT# writes to the printer).

dress ENTER

PRINT#

Commas and semicolons react the same way with PRINT# that they
do with PRINT statements. If you were to write the following
program:

10 M=10.7
20 N=20.2

to a disk using PRINT##A.B (as opposed to PRINT##A:B), then
you would put this on the disk: 10.7  20.2

See the extra spaces? Those are 13 blanks that BASIC writes to
the disk.

dress ENTER

PRINT##A.B

The above command puts this on disk: 10.7  20.2

If you use a semicolon, like this, PRINT##A:B then you get:

10.7  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?

DO OPEN*2, 'NEW2DS'
DO PRINT#2, P#1
DO CLOSE :

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#inc.

Press ENTER.
**** Listing of Program "LESSON4/TXT" ****

10 OPEN "I",1,"TEST"
20 INPUT#1,NS
30 PRINT NS
40 CLOSE 1

The format for the INPUT# statement is similar to the PRINT# statement. It is: INPUT# (buffer#)(variable1)(variable2)(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 the 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 any PRINT# statements were used.

C How 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
### Listing of Program 'LESSON4/TXT'

**INPUT# & EOD**

If we are inputting STRING data our variable is a STRING such as INPUT#1.W#. 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 EOD marker (more about EOD 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 EOD marker.

Press ENTER.

**INPUT# and EOD**

Here is an important concept to understand about how INPUT# works when you use STRINGS. IF YOU PRINT A STRING ON 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 "MC 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.

50 INPUT#1, A$
60 INPUT#1, N

This will solve the problem of mixing strings and numbers on disk.

press ENTER

10 OPEN "C:\1\**TEST**
20 INPUT#1,N$
30 PRINT#1, 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 "C:\1\**TEST**
20 INPUT#1,N$
30 PRINT#1, 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 29 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
to put 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"TEST"
20 IF EOF() THEN GOTO 30
30 INPUT$100
40 PRINT N$
50 GOTO 20
60 PRINT"END OF FILE ENCOUNTERED"
70 CLOSE 1

.............more of the program or an END statement

Press ENTER

10 OPEN"TEST"
20 IF EOF() THEN GOTO 30
30 INPUT$100
40 PRINT N$]
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
**** Listing of Program `LESSON4.TXT` ****

would pass to line 20. 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'd get an ERROR
press ENTER

INPUT# and EOF

The format for the EOF statement is:

EOF:buffer #1:

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"TEST"
20 IF EOF() THEN STOP
30 INPUT#1,W
40 PRINT W
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 tries to re-OPEN the
file that hasn't been CLOSED. Correct answer is C

press ENTER
What is wrong with this program?

1) OPEN IN... "TEST"
2) IF EOF THEN STOP
3) INPUT #1, N
4) PRINT N
5) GOTO 2

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.

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 E00  : Clears string space
20 INPUT"Input filename":F1$   
30 INPUT"Output filename":F2$   
40 OPEN#1,FILE:1  : OPENS the INPUT file
50 OPEN#2,FILE:2  : OPENS the OUTPUT file
60 IF EOF#1 GOTO 110  : Checks for end of file on file #1
70 INPUT#1,F1$  : INPUT's data from file #1
80 PRINT#2,F2$  : PRINT's data to the new file
90 GOTO 80   : Goes back for more from file #1
100 INPUT"ENTER DATA enter 999 to stop":C$  
110 IF C$ = "999" GOTO 140
120 PRINT#2,F2$  : Updates new file with your data
130 GOTO 100   : Goes back until line 11: 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

dress 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 questions). 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 you and things together for you. GOOD LUCK!

dress 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  LIST
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

ERROR - 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.

To transfer data from your file to a disk, the computer must first send the data to a buffer where it is processed. This 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 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 E. The filename is B5ZDAT

A. 10 OPEN 'E', 1, 'B5ZDAT'
B. 20 OPEN 'E'. 1, 'B5ZDAT'
C. 30 OPEN 'E'. 1. 'B5ZDAT'
D. 40 OPEN 'E'. 1, 'B5ZDAT'
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 file when it is executed):  

1) OPEN"F:1,"TEST"  
2) IF EOF(1) THEN END  
3) INPUT A$  
4) PRINT A$  
50 GOTO 10  

A: Nothing  
B: A "File already open" ERROR will be generated  
C: The wrong file name is used  
D: Line 20 is illegal  

ENTER the letter opposite the correct answer:  

CORRECT  
press ENTER  

"The CLOSE word can 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:  

CORRECT  
press ENTER  

what is wrong with the following program?  

10 OPEN"F:1."TEST"  
20 PRINT A$11;11;20;30  

**** Listing of Program 'LESSON4.TXT' ****  

TO CLOSE

A Nothing 
B Wrong file mode is used 
C The CLOSE statement is invalid 
D The PRINT#1 statement should be INPUT#

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 #4.

ENTER the command now: to INPUT#3, #4

CORRECT

press ENTER

What is wrong with the following program?

10 IF EOF THEN GOTO 40  
20 INPUT #1  
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

412
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  "D"

Which? A

You are now qualified to go to LESSON 5.

Do you wish a homework assignment for? Y/N

Press 'Y' for Y

Press 'N' for N

SYSTEM**EXIT** ***00
TRSDOS Ready

BASIC

This is a tutorial for TRSDOS Version 3.

Copyright © 1982 B. Microsoft. Licensed to Taney Corporation.
All rights reserved.

Read.

run "Testert"

---

LESSON: BASIC 5
VERSION: 1 AUGUST 80

TIME REQUIRED TO COMPLETE LESSON: About one hour

AUTHOR: Capt Gary J. Creegan
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
2. Subroutines
3. Nested Subroutines
4. ON ERROR
5. STOP / RETURN

1) Return to this part of the lesson...
2) I want to review selected areas.
3) I want to go to the second part.
4) I want to return to the "end."
INTRODUCTION

In the last episode of our computerized book, we assigned you a homework problem that dealt with disk I/O, disk output. It included most of the concepts that we have been studying. Now we are on the downhill part of the course. That is 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 you do not have to memorize them. We 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, set 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 on

B Review this section Again

dress the letter opposite the correct answer and press ENTER

---
SUBROUTINES

SUBROUTINES

We use the word "ROUTINE" to describe the statements in the body of a program. "SUBROUTINE" is used to describe a block of code that was built, attached to the main program, and used to perform a mini-task that, for some reason, needs special attention.

Lusiv... we use subroutines to do tasks which we perform often in our program such as printing out a menu after each module of a large program is finished by a student. That we... we do.. have to write the menu program once and call on it when we need it.

Press ENTER.

1. Would we want to use a subroutine?

2. To those parts of the program that are used often.

3. To do things the normal routines

INPUT the letter opposite the correct answer.

0093 - this was supposed to be an easy question to answer.
In...... we screwed up somehow. We will send you back to the reading of this part. "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 block of code that was built, attached to the main program, and used to perform a mini-task that, for some reason, needs special attention.
SUBRoutines

Subroutines are main-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.

1: PRINT "A CMI program is a computer assisted instruction."
2: PRINT "This program that teaches students."
3: IF "YES" = "YES" THEN GOTO 10 ; If returns from sub with choice
4: "......END"
5: "NO" = "NO" THEN GOTO 10 ; If returns from sub with choice
6: "......END"
7: END
8: The subroutine follows
9: END
10: "INPUT Do you want to continue or review what you've just done?"
11: "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?

1. The beginning of the program
2. The calling statement
3. The statement after the calling statement
4. None of the above
**** Listing of Program "LESSON14A" ****

Enter the letter opposite the correct answer:

CORRECT = Super 100 .... Now we can go on

Press ENTER:

Which do you wish to do?

A. Continue on
B. Review this section again

t... the letter opposite the correct answer and press ENTER. A

SOSUB & RETURN

The set of statements that you use to implement a subroutine is made of the SOSUB and RETURN words. The SOSUB 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 SOSUB 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:

(1) print A CRI program is a computer assisted instruction
(2) print "tutorial that teaches students."
(3) SOSUB 1000 \* \* \* \* CDR use sub. after every major part
(5) IF # = 10 THEN goto 11 \* \* TA returns from sub with choice
(6) \* \* \* ETC
(8) SOSUB 100 \* \* \* CDR use sub. after every major part
(11) END \* \* you must ensure program doesn't goto sub by accident
(12) REM
(13) REM the subroutine follows
(14) REM

473
+++ Listing of Program "LESSON7.TXT" +++

1000 INPUT "Do you want to continue or review ?#Con B=Rev "7#
1010 RETURN

Line 10 and line 20 call the subroutine, and line 10410 RETURNS to the appropriate statement. press ENTER

10 print "A QAL program is a computer assisted instruction"
20 print "Program that teaches students..."
30 GOSUB 10000 "-""-""- LOCK sub. after every major part
40 IF T1 = "S" THEN GOTO 10: T1 returns from sub with choice
50 "... ETC
30 GOSUB 10000 "-""-""- LOCK sub. after every major part
60 END If you must ensure program doesn't goto sub by accident
70 REM
80 REM the subroutine follows
90 REM
1000 INPUT "Do you want to continue or review ?#Con B=Rev "7#
1010 RETURN

If line 10 calls the subroutine, what line gets control after
the RETURN statement... ENTER the correct line number? 51

CORRECT - You are getting 33000000000000000000

press ENTER

GOSUB 10 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 an 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
110. See line 120 and put the space between all terms.

Enter the command now: 100 GOSUB 1100

CORRECT.

press ENTER

Which do you wish to do?

- Continue or
- Review this section again

press the letter opposite the correct answer and press ENTER
NESTED SUBROUTINES

What do you think the output of the following is?

```
10 PRINT "Main Program"
20 GOSUB 100
30 GOSUB 200
40 END
10 PRINT "Subroutine One"
20 GOSUB 200
30 RETURN
20 PRINT "Subroutine Two"
30 RETURN
```

cross ENTER for the answer

The answer is:
```
10 PRINT "Main Program"
20 GOSUB 100
30 GOSUB 200
40 END
10 PRINT "Subroutine One"
20 GOSUB 200
30 RETURN
20 PRINT "Subroutine Two"
30 RETURN
```

Main Program
Subroutine One
Subroutine Two
Subroutine Two

cross ENTER"
***** Listing of Program 'LESSONS/TAT' *****

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
110 RETURN

"We put the line numbers as they would be executed, putting a space between each number. (ie 10 20 30 40) Enter your answer"?

110:20:40:30 - the correct answer is (110 100 20 20 100 210 30)

"That was a rough one. You may want to review this part

Press ENTER?

Which do you wish to do?"

A Continue on
S Review this section again

Press the letter opposite the correct answer and press ENTER" A
Reember the ON GOTO statement from our previous lessons?

There's a command very similar to the 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"  
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"  
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
600 FOR I = 1 TO 50
610 PRINT I*I*I
620 NEXT I
700 RETURN
800 FOR K = 1 TO 50
900 PRINT A, K*K*K
1000 NEXT K
1100 RETURN
2030 RETURN

If you ENTER a 1, then subroutine 1000 is used. Which subroutine 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

c. 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
2. Functions Overview
3. Library Functions
4. User Defined Functions
5. DEF Statements
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 categories 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 common, 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

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(expr) - gives absolute value 7) LOG(expr) - gives LOG(e)
2. ATRN(expr) - arctangent in radians 8) RND(expr) - random numbers
3. COS(expr) - returns cosine of expr 9) SGN(expr) - sign of expr
4. EXP(expr) - natural exponential 10) SIV(expr) - sine of expr
5. FIX(expr) - gives integer of expr 11) SQRT(expr) - square root
6. INT(expr) - gives integer of expr 12) TAN(expr) - tangent(expr)

expr = an appropriate numeric expression - remember, you can't use -M for some of the functions (SGN(-1) bombs)

Specific examples of these functions will be given later. Or you can 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 can be used at any time in your program.

4. TRUE
5. FALSE

ENTER the letter opposite the correct answer: A
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

Which do you wish to do?

A Continue on
B Review this section again

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

Library Functions
1) ABS(exp) - gives absolute value 7) LOG10(exp) - gives LOG10(exp)
2) ATN(exp) - arctangent in radians 8) RAND(n) - 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) FINT(exp) - gives integer of exp 11) SQRT(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 AN: number for base of the functions (SQRT(1) works!)

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 the statement - PRINT LOG(10)
The example of the log of 10 is exactly how we implement the library functions. Here is another example:

10 INPUT "Enter the number you want the square root of": N
20 PRINT SQRT(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

1: INPUT "Enter the number you want the square root of": N
2: PRINT SQUARERT(N) 
3: GOTO 11

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.
pres ENTER for an example?
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 as a positive number, and then SQRT(ABS(N)) will give the square root.

You can use just about any valid numeric expression for the value part of any library function.

press ENTER

Is the following statement valid

10 PRINT LOGIC

HINT: If you are not sure, try it on a calculator

A Yes, it is valid
B No, it 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 SQRT ABS(N)
30 GOTO 10

The expression ABS(N) is evaluated first, then the outside expression - SQRT(----) - is evaluated next. Remember from the first lesson that 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 parentheses is used, then the expression within the inner-most set of parentheses 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: SQRT(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

What is the function that gives the arctangent of A? ATN(A)

WRONG - the correct answer is ATN(A)

What is the function that gives the sine of .5? SIN(.5)

CORRECT

What is the function to give a random number between 0 & 1? RND

WRONG - the correct answer is RND(10)

Which do you wish to do? A. Continue on
P. 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 FNRI = INT(RAND*10) + 1
20 DEF FNW(A,B) = A * B/2 + (A - B)
30 X = FNRI
40 PRINT X
50 Y = FNW(1.4)
60 PRINT Y
70 END

403
There are two user functions in this program. They are defined in lines 10 and 20, and called on in lines 50 and 50. We will cover this program in greater detail in the next section.

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

DEF Statement
The DEF statement is used to define a user function and it has the following format:

```
LINE #: DEF FN function name(=vars) = expr
```

You must never use the DEF statement in the IMMEDIATE mode, you always have to have a line number. In CPM and Commodore systems, you must separate the terms DEF and FN by one space. In TRS-80s you don't have to. The function 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 expr is the calculation that you wish the function to do.

press ENTER.
Jet out your BASIC manual and look up DEF. It will show you examples similar to this. Line 10 defines a function named R: 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 FNRI = INT(RND(1) * 10))
(20 DEF FNW(4,3) = A * B + (A - B))
(30 X = FNRI)
(40 PRINT X)
(50 Y = FNW(1,4))
(60 PRINT Y)
(70 END)

Line 10 calls on the function defined in line 10. When it does, it 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, AN: valid variable can be passed to a function, even a string.

Press ENTER.
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 DEFned 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.

Press ENTER.

The two values that are DEFINed as A and B will be set equal to the corresponding values of 3 and 4. A will equal 3 and B will equal 4. The DEF statement will then use these numbers to calculate the factor part of the statement, and then Y will be made equal to this value. Finally, the value will be printed in line 50. You may wish to copy a program like this and experiment with it to see how it works.

Press ENTER.
***** Listing of Program 'LESSONS.7AT' *****

40 PRINT x
50 y = FNM(x, 4)
60 PRINT y
70 END

What value would be passed to y 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 FNM(v, #) = LEN(v)#
20 v# = "This is a function"
30 x = FNM(v, 4)
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 "FM.var ". That is the only way to get to your
user function.

Press ENTER

10 DEF FNM(v, #) = LEN v#
20 v# = "This is a function"
30 x = FNM(v, 4)
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 size 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?

<table>
<thead>
<tr>
<th>A</th>
<th>DEF FNSB.A.B. = A * B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>yes. To call on it simply put FNSB somewhere in your program</td>
</tr>
<tr>
<td>B</td>
<td>no. The terms A and B are not defined and will cause errors</td>
</tr>
<tr>
<td>C</td>
<td>No, B8 is not a valid variable for this position</td>
</tr>
<tr>
<td>D</td>
<td>No, the syntax is good, but the beat is poor. You can't</td>
</tr>
<tr>
<td></td>
<td>dance to it.</td>
</tr>
</tbody>
</table>

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

dress 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 the 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 successfully complete the test, you can go on to the last lesson.

GOOD LUCK

press ENTER to continue.

Line in the order in which the following lines will be executed. Leave one space between each line number - i.e. if the execution sequence is ten, twenty and thirty, then type in 10 20 30.

10 20 30
***** Listing of Program 'LESSONS.FAT' *****

10 PRINT "DONE"
20 END
3000 RETURN

Enter the sequence now? 10 1980 20 30
CORRECT
press ENTER?

What will happen when you input then number 4 to the following program?

10 INPUT A
10 ON A GOSUB 1000,2000,3000
10 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? D
CORRECT
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?

END
What is the term you would use to define a FUNCTION.

hint: It’s two words. DON’T put in a variable. Leave a space between the two words.

ENTER your answer now for DEF FN

CORRECT

press ENTER

10 DEF FNX = 10 + 2
20 Y = FNX

What is the term that would complete line 20 if we wished to call on the user function in line 10?
***** Listing of Program 'LESSON/T1' *****

ENTER your answer now: FNX

CORRECT
press ENTER?

Which of the following functions is valid?

A DEF FN(T.1) = 2 * M/2 + AS
B DEF FN(1.3.4) = 2 * 2 + 4 + M
C DEF FN(R,A): = LEN(R)+A + 1
D DEF FN = SQRT(-4)
E DEF FN = 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
Do you want your score recorded on a permanent file?
A: YES
B: NO
Which? B

You are now qualified to go to LESSON b.
you may return to the MENU or receive your homework.

Do you want your homework assignment (Y/N)?
Break in 30 sec.
Read,
SYSTEM*RESET *CD
Lesson 5

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

1: Introduction
2: String Assignment
3: String ARRAY's
4: Concatenation
5: String Functions

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.
"tis our last lesson. Congratulations. 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 any 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 VS = "Another way is to put it in a variable and print it"
30 PRINT VS

Line 10 prints the string data immediately, and line 20 loads the variable VS with the data. VS can then be printed whenever we want it.

press ENTER

INTRODUCTION

10 PRINT "This is one way to designate a string"
20 VS = "And this is another"
30 PRINT
40 PRINT VS
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 lowercase 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 $\text{A}$ is blank)

10 READ $\text{A}$
20 INPUT $\text{A}$
30 LET $\text{A} = \text{B}$
40 $\text{A} = "\text{NOW IS THE TIME}"$
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.

cross ENTER?

String Assignment

10 READ $\text{A}$, $\text{A}$
20 PRINT $\text{A}$
30 $\text{A} = \text{B}$
40 PRINT $\text{A}$
50 DATA "THE GRINCH", "IS COMING"

RUN

THE GRINCH
IS COMING

Notice that $\text{A}$ was converted to $\text{B}$, and all the data was printed out by using just $\text{A}$ in print statements.

cross 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 NNS and use line number 10 as your statement number.

Put one blank between terms.

ENTER your answer now?

WRONG - the correct answer is 10 NNS = "NOW IS THE TIME"
(you could have also answered 10 NNS = "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 *, 1, "TEST"
20 IF EOF(1) THEN END
30 INPUT 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
Listing of Program 'LESSON.TXT' 07/11/83 - 01:17:40

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.
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"...TEST"
20 IF EOF(1) THEN END
30 INPUT@,A$
40 PRINT A$
50 END 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?
1. Continue on
2. Review this section again

press the letter opposite the correct answer and press ENTER*
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$$(100)
30 i = 0
40 i = i + 1
50 INPUT "ENTER up to 99 strings, ENTER "END" to stop":A$$(i)
60 IF A$$(i) = "END" THEN END
70 GOTO 40

press ENTER

Notice the CLEAR statement. Remember that you normally have only 1.25 - 1.99 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$$(100)
30 i = 0
40 i = i + 1
50 INPUT "ENTER up to 99 strings, ENTER "END" to stop":A$$(i)
60 IF A$$(i) = "END" THEN END
70 GOTO 40

Notice the CLEAR statement.
### Listing of Program "LESSON5.TXT" ####

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 $ seems acting as a counter to reference the proper packet 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 terminate all numeric variables
D. To clear the screen
E. To help the programmer understand more clearly

Enter the letter opposite the correct answer: 

Correct?

Press ENTER?

### String ARRAYS

10 CLEAR 2000
20 DIM A$(1991)
30 x = 0
40 i = x + 1
50 INPUT "ENTER up to 99 strings. ENTER "END" to stop":A$(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 line of a string array?

If you don't use the DIM statement... type "answer now"
**** Listing of Program 'LESSON4.TXT' ****

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
simple stackon array A$(x)?

4  PRINT A$(x)
3  PRINT A$(5)
0  PRINT A$
5  READ A$(x)

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

which do you wish to do?
4  Continue on
3  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:
1) A$ = "where"
2) B$ = "time"
3) C$ = B$ + A$
4) PRINT C$  (you could have said 'PRINT B$ + A'$ too.
RUN

455
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
C. DANCE
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

correct 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.
***** Listing of Program "LESSON6.TXT" *****

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); 5) LEN(string);
2) CHR#(char); 6) MID$(string, position, length);
3) FRL$(string); 7) RIGHT$(string, length);
4) INEAV; 8) LEFT$(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 go 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
String Functions

10 A$ = "aachinringers"
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 STANDARAG 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 '97 decimal. When your computer writes data files to disk, it usually writes them in ASCII code, one letter at a time. This function is useful when you are trying to convert characters to their number equivalent.

cress ENTER

String Functions

PRINT CHR$ '77
CHR$ e.g. returns the opposite of the ASCII string function. It returns a character equivalent of decimal '77'. Which is 'A'.

cress ENTER

What is the output of the following program?

10 PRINT ASC 'A'
20 PRINT CHR$ 77
30 PRINT ASC 'A'
40 PRINT ASC 'A'
50
***** Listing of Program 'LESSONS/TAT' *****

C = 1
D = 3
C

ENTER the letter opposite the correct answer: A
CORRECT - GREAT!
cress ENTER

String Functions

1) A$ = "My aching fingers"
2) PRINT LEN(A$)

LEN(string) is a function that returns the length of the string that is in parentheses. In this case it should return 11.

What is the value of LEN("TEST")?
A: 4
B: 5
C: 1
D: 0

ENTER the letter opposite the correct answer: A
which - the correct answer is 4
press ENTER

Now you have to do some work for yourself. What will be the output of the following program?

I) A$ = "My aching fingers"
2) PRINT LEFT$(A$,3)
3) B = 1
C: 1
D: 0

C: My aching
***** Listing of Program "LESSON/MIT" ***** 07/11/80 - 01:17:40

0  Nothing will be output
ENTER the letter opposite the correct answer? A
CORRECT
press ENTER

10 A$ = "My aching fingers"
20 PRINT MID$(A$,4,4)
ENTER the output of this program aching
CORRECT - I'm glad to see you use the book!
pres ENTER

10 A$ = "My aching fingers"
20 PRINT RIGHT$(A$,4)
ENTER the output of this program? @Berks
CORRECT - good job
press ENTER

String Functions

Here's an interesting function. It's called INKEY$ and it
strokes 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
Let's keep running until you press the 'B' 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. nDelete
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 rewrite 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 have 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.

```plaintext
10 FOR I = s**to77 : PRINT I: NEXT I
```

The colon forms 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. Obvious... there are several errors in the line.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```

Starting

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.

```
10 FOR I = s**to77 : PRINT I: NEXT I
```
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 = KeT57 : PRINT #: NEXT 2

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.

nSPACEBAR, nSearch!

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 = KeT57 : PRINT #: NEXT 2

EDIT 10

10 FOR X = KeT57 : PRINT #: NEXT 2

4-4
**** Listing of Program "LESSONS/TA1" ****

2000 "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 nSEARCH

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 is, you won't have to 
move all the way to your spacebar to get to the 20th character 
of a long line. All you do is press the number of characters 
you want to "spaceover" over and then press the spacebar. 
In this example, you are in the edit mode and you want to go 
over 9 new spaces.

Press a number (up to 350), and then press the spacebar.

1. FOR i = 1 TO 100 PRINT i : NEXT i
2. EDIT 10
3. FOR i
4. END

"NBA. that was tricky... if you want to do it again, ENTER a 'y'
else ENTER an 'n' instead.

nSPACEBAR nSEARCH

If you entered a bigger number that your line length, then 
the computer just defaulted to the previous 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.

Dr. we assume 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 'nSEARCH' feature.
***** Listing of Program 'LESSON/Txt' *****

press ENTER

*SPACEBAR - Search:

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 beeps 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 'I'. All you do is press '9' and then press 'I'. The editor recognizes upper and lower case, so be sure the case is right. So sheed and do it now. You may search for any letter, just type 5 first, then type the letter.

```
1: FOR I = @to99; PRINT I; NEXT I
EDIT 10
1: FOR I = $to99; PRINT I; NEXT
```

Now, another quick one... ENTER '9' to do again, else ENTER an 'N' 'I'

*SPACEBAR - Search:

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 'i' in the title 'i3 search'. It simply means that you can search for the i'th occurrence of the specified character.

For example, if we were in the EDIT mode for line 10 and we wished to find the second occurrence of the letter 'i', then we must type 'i29' and the cursor would stop over to the 9'th right i.
Here is an example of the `SEARCH` feature.

1. Press `?` to search for a character.
2. Type a character to search for.
3. Press `Enter` to search.

What would you type to find the 2nd occurrence of the character "\" in a line you were editing, unless you are already in the EDIT mode?

Correct = `OPEN`
Press `Enter`.

What do you wish to do?
- Continue or
- Review this section again
Press the letter corresponding to the correct answer and press `Enter`.

C. Review.
For example, if you are in the EDIT mode for line 11 and you want to delete the NEXT character, then just press 1 and the character will be enclosed in exclamation marks. The exclamation marks indicate that if you don't change things, then the "next line" will not have the character in it.

Press ENTER

An example of the Delete command would be:

```plaintext
1. FOR I = 1024 PRINT "NEXT I"
2. EDIT 11
   "Now say we want to delete the 'K' character. All we do is put the cursor to the left of the two characters and press 1 twice"
3. FOR I = 1024 PRINT "NEXT I"
   "Then it would look like this. Then we would press ENTER and we would have this"

Ex: This example and read the appropriate paragraph on your helix...

Press ENTER
```

"Delete"

```plaintext
1. FOR I = 1024 PRINT "NEXT I"
```

Now it is your turn. You have to EDIT the above line so that the characters 'K' are deleted. You just delete them one at a time. As we showed you in the previous example. First type the appropriate command for setting line 11. Then move the exclamation over to the appropriate space. Then delete the two preceding characters, then press ENTER. Use lower case.
***** Listing of Program "LESSON7.TXT" *****

ENTER the first command now

WHERE = you should type in EDIT 10 first

ENTER the first command now EDIT 10

10 FOR I = " Sage" PRINT I; NEXT I

The line in the computer's memory would now look like this:

10 FOR I = " Sage": PRINT I; NEXT I

THEN ... If you want to do it again, press 'r' else press 'n'

ENTER your choice now 'Y' or 'N'

"Delete"

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,
If you delete two what you have done, the editor is very
forgiving. It usually leaves you something, even when you have a
lot done.

As you can see below, the 'r' symbol in the title indicates
that you really are 'r' characters at a time. If you press a
"sage" and press b, then that is how many characters will be
deleted.

press ENTER

10 FOR I = " Sage": PRINT I; NEXT I

If we wanted to delete all at once, we could position the
cursor to just before the 's' and type in 10 and both
characters would appear like this: ' ",

Write the statement 'r' times, in this case, are around both
characters. When you set through here, practice with some
***** Listing of Program [REDACTED] *****

Lines you have already done now, you will see that this
command can be very handy.

Press ENTER.

which do you wish to do?
- Continue on
- Review this section again.

Press the letter opposite the correct answer and press ENTER.

I tend 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
were pressed ENTER.

First, you go to EDIT mode, then you press k, when you do, you
will see the whole line displayed, and you can edit anything
up to the END or it. Try it now. First, type the command to
set into EDIT mode for line 1, then press k, then type in
where you want that move allow a maximum of 249
characters. Then press ENTER. Do it now.

ENTER the first command? "EDIT 1"

1. FOR k = 1 TO 10 PRINT "MREN 1"

Our new line, which was the sum of the old line 115 is the
data you store on the flow chart:

The = PRINT "MREN 1"

Just press k, you will find yourself using this command the most.

As usual, we protected you from screwing up your file, try.

allowing you to execute the 'l' (insert) 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 wide out the end of the line you are editing, and replace the old data with the 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

dress the letter opposite the correct answer and press ENTER. A

# Change 1: insert

Now for the rest of this half -- we are going to change the first error in our trial statement, and then insert some correct figures.

FOR 1 = LEFT PRINT : NEXT I

Give us our line with the "set" characters missing -- we deleted them in the add delete section). Let's say we want to change "LEFT" to "RIGHT" and "PRINT" to "PRINT".

In addition, we want the value of I to start at 1 in the FOR "SET" loop, i.e., we want to insert a 1 just before the tot

dress ENTER to start our EDITING!

# Change 1: insert

FOR 1 = LEFT PRINT : NEXT I

which we'll change "LEFT" to "RIGHT".
To use the \texttt{CHANGE} command, get into the EDIT mode and position the cursor to JUST BEFORE the character to be changed. Then press \texttt{M} and press the new character, then press \texttt{ENTER}. Try it now. Set to EDIT mode, SPACE over to just before the \texttt{Y} and type in \texttt{Y'}, then type in an \texttt{X} (we want to switch \texttt{Y} with \texttt{X}), and finally press \texttt{ENTER} - use capitals.

\texttt{ENTER} the first command at the bottom of the next line
\texttt{10 FOR } \texttt{I} \texttt{= TO I : PRINT } \texttt{Y} \texttt{: NEXT } \texttt{I}
\texttt{EDIT } \texttt{I}
\texttt{10 FOR } \texttt{I} \texttt{= TO I : PRINT } \texttt{X} \texttt{: NEXT } \texttt{I}

How's that for class? Remember, we protected you from mistakes. The real editor will do whatever you tell it, even if it is wrong. But you knew that, didn't you?

\texttt{ENTER} a 'Y' if you want to do this again, else \texttt{ENTER} an 'N' \\

\texttt{CHANGE & INSERT}

As with the other commands, the 'N' in \texttt{CHANGE} designates how many characters are affected by the command. If you want to change 10 characters, then you would type \texttt{"10C"} in the EDIT mode, and you would then \texttt{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 \texttt{"\textsc{t}"}?

\texttt{ENTER} your answer now: \texttt{4C\textsc{t}}
\texttt{EDIT} now you have the idea
\texttt{press \texttt{ENTER}}

\texttt{CHANGE & INSERT}

\texttt{10 FOR } \texttt{I} \texttt{= TO I : PRINT } \texttt{Y} \texttt{: NEXT } \texttt{I}
we would change the "I" to an "X" in the same way, but to
save time, we'll emuere some BASIC and change it now so we can
get to the Insert: command. 

"READY" **********CHARGE

... FOR X. There, it's changed now. Look below.

10 FOR X = to77: PRINT X: NEXT X

How would you like to have THE editor at your command?

we'll now get to the Insert: command. Remember, we want to
insert a space just before the "to77".

press ENTER for the Insert example:

INSERT: and Insert:

11 FOR X = to77: PRINT X: NEXT X

To use the Insert: command, you first get into the EDIT mode
and then place the cursor to just before the character you want
(or 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 Insert:. 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:

Presser, first type EDIT 10; then space over to just before
the "to77", then type "I", then type a 1, then type ENTER.

12 FOR X = to77: PRINT X: NEXT X

ENTER the first command: EDIT 10

13 FOR X = to77: PRINT X: NEXT X
***** Listing of Program 'LESSON:74' *****

ENTER a 'Y' to do this again, else ENTER an 'N' or 'N

...change, & [insert]

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 our nC or I command to fix up the line
but I feel pretty magical today, so I want to do it.

&zzzzzzzzzzzzzzzzzzzzzzzzzzssenedn .........CRANBERRY ... tinkle
[FOREGROUND]TWEETFLUSH()****slurrrrrrrrrrr............goodbye'

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

cross 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', 'R'

Send to test. Please wait one second.

FINAL TEST : Lesson 6

474
This test consists of 10 questions. You must get 70 percent or more 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

WARNING - the correct answer is H  
1 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 t = 1 to 20
20 A$(t) = "0"
30 NEXT t

- The array is not dimensioned properly
- You cannot address a single dimensioned array with a loop
- The '0' should not be enclosed in quotes
- 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$ = "44"; B$ = A$ ; B$ = LEFT$(A$,1); PRINT B$
20 B$ = A$ + B$

ENTER your answer EXACTLY as it would appear

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

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 occurrence of the letter R in your line.

What is the command you would use?

A  ZR
B  ZCR
C  ZRA
D  ZCR
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?  ;

CORRECT

press ENTER?
Assume you are in the EDCI code.

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 EDCI. What should you do?

A continue
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

Obvious... we have failed to communicate. I'm reporting you to the FBI for tax evasion and mail fraud.

press ENTER

Assume you have just finished a course in computer assisted instruction in EDCI. What should you do?

A continue
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 8.

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?

- YES
- 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 #230
Ready
svetsel-setel $00
APPENDIX D

HOMEWORK MODULES
# Table of Contents

<table>
<thead>
<tr>
<th>Homework for Lesson 2</th>
<th>483</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework for Lesson 3</td>
<td>485</td>
</tr>
<tr>
<td>Homework for Lesson 4</td>
<td>487</td>
</tr>
<tr>
<td>Homework for Lesson 4</td>
<td>489</td>
</tr>
</tbody>
</table>
Listing of Program "HWZ" ******

1900 GOSUB 1520
1910 PRINT "Your lesson 2 assignment is to write a program that requests the"
1920 PRINT "user to INPUT a number. The program should check the number,"
1930 PRINT "against numbers in a data statement to see if there is a"
1940 PRINT "match. If there is a match, then the program should tell"
1950 PRINT "the user. Use the RESTORE and GOTO statements to keep the"
1960 PRINT "program running. Save the program to disk."
1970 PRINT
1980 PRINT "You will use much of what we have learned when you write this"
1990 PRINT "program. If you feel a little confused by the directions,"
2000 PRINT "you can get an example of a program that satisfies the"
2010 PRINT "requirements by turning on your printer and selecting the"
2020 PRINT "LIST PROGRAM TO PRINTER option below."  
2030 PRINT
2040 PRINT "Which do you want to do?"
2050 PRINT "A) RETURN TO MENU"
2060 PRINT "B) LIST PROGRAM TO PRINTER"
2070 PRINT "C) RUN THE PROGRAM"
2080 PRINT
2090 INPUT "Enter your choice: "
2100 IF T$ = "A" THEN RUN "MENU"
2110 IF T$ = "B" THEN PRINT; PRINT "Be sure to type in RUN when the printer is done."; LIST 123-1500
2120 IF T$ = "C" THEN GOSUB 1520; PRINT; PRINT "Be sure to type RUN when program ends."; PRINT; INPUT "Press ENTER to start: "; RUN 1230
2130 GOTO 1140
2140 REM "his program asks for a number, compares the number to a"
2150 REM data list, and then tells the user if the correct number"
2160 REM was selected. It is not the only way to do the problem.
2170 REM
2180 REM Next line resets the data statement (see lesson 2, part 2)
2190 RESTORE
2200 REM Now we ask for the number. I use 1 to 20 to keep it simple.
2210 INPUT; "Enter a number between 1 and 20: ";
2220 IF N = -1 THEN GOTO 1500
2230 REM Now we read a data point to see if we match (See lesson 2, part 2)
2240 READ 0
2250 DATA 1,2,7,10,15,19,20
2260 REM Above DATA is arbitrary; I just made a few points up
2270 REM Now we compare the data point with the inputted number (see part 2)
2280 IF N = 0 THEN PRINT "MATCH - There is a data point that equals:"
2290 IF N = 0 THEN PRINT "";
2300 IF N = 0 THEN PRINT "";
2310 IF N = 0 THEN PRINT "";
2320 IF N = 0 THEN PRINT "";
2330 IF N = 0 THEN PRINT "";
2340 IF N = 0 THEN PRINT "";
2350 IF N = 0 THEN PRINT "";
2360 IF N = 0 THEN PRINT "";
2370 IF N = 0 THEN PRINT "";
2380 IF N = 0 THEN PRINT "";
2390 IF N = 0 THEN PRINT "";
2400 IF N = 0 THEN PRINT "";
2410 IF N = 0 THEN PRINT "";
2420 IF N = 0 THEN PRINT "";
2430 IF N = 0 THEN PRINT "";
2440 IF N = 0 THEN PRINT "";
2450 IF N = 0 THEN PRINT "";
2460 IF N = 0 THEN PRINT "";
2470 IF N = 0 THEN PRINT "";
2480 IF N = 0 THEN PRINT "";
2490 IF N = 0 THEN PRINT "";
2500 IF N = 0 THEN PRINT "";
2510 IF N = 0 THEN PRINT "";
2520 IF N = 0 THEN PRINT "";
2530 IF N = 0 THEN PRINT "";
2540 IF N = 0 THEN PRINT "";
2550 IF N = 0 THEN PRINT "";
2560 IF N = 0 THEN PRINT "";
2570 IF N = 0 THEN PRINT "";
2580 IF N = 0 THEN PRINT "";
2590 IF N = 0 THEN PRINT "";
2600 IF N = 0 THEN PRINT "";
2610 IF N = 0 THEN PRINT "";
2620 IF N = 0 THEN PRINT "";
2630 IF N = 0 THEN PRINT "";
2640 IF N = 0 THEN PRINT "";
2650 IF N = 0 THEN PRINT "";
2660 IF N = 0 THEN PRINT "";
2670 IF N = 0 THEN PRINT "";
2680 IF N = 0 THEN PRINT "";
2690 IF N = 0 THEN PRINT "";
2700 IF N = 0 THEN PRINT "";
2710 IF N = 0 THEN PRINT "";
2720 IF N = 0 THEN PRINT "";
2730 IF N = 0 THEN PRINT "";
2740 IF N = 0 THEN PRINT "";
2750 IF N = 0 THEN PRINT "";
2760 IF N = 0 THEN PRINT "";
2770 IF N = 0 THEN PRINT "";
2780 IF N = 0 THEN PRINT "";
2790 IF N = 0 THEN PRINT "";
2800 IF N = 0 THEN PRINT "";
2810 IF N = 0 THEN PRINT "";
2820 IF N = 0 THEN PRINT "";
2830 IF N = 0 THEN PRINT "";
2840 IF N = 0 THEN PRINT "";
2850 IF N = 0 THEN PRINT "";
2860 IF N = 0 THEN PRINT "";
2870 IF N = 0 THEN PRINT "";
2880 IF N = 0 THEN PRINT "";
2890 IF N = 0 THEN PRINT "";
2900 IF N = 0 THEN PRINT "";
2910 IF N = 0 THEN PRINT "";
2920 IF N = 0 THEN PRINT "";
2930 IF N = 0 THEN PRINT "";
2940 IF N = 0 THEN PRINT "";
2950 IF N = 0 THEN PRINT "";
2960 IF N = 0 THEN PRINT "";
2970 IF N = 0 THEN PRINT "";
2980 IF N = 0 THEN PRINT "";
2990 IF N = 0 THEN PRINT "";
3000 IF N = 0 THEN PRINT "";
3010 IF N = 0 THEN PRINT "";
3020 IF N = 0 THEN PRINT "";
3030 IF N = 0 THEN PRINT "";
3040 IF N = 0 THEN PRINT "";
3050 IF N = 0 THEN PRINT "";
3060 IF N = 0 THEN PRINT "";
3070 IF N = 0 THEN PRINT "";
3080 IF N = 0 THEN PRINT "";
3090 IF N = 0 THEN PRINT "";
3100 IF N = 0 THEN PRINT "";
3110 IF N = 0 THEN PRINT "";
3120 IF N = 0 THEN PRINT "";
3130 IF N = 0 THEN PRINT "";
3140 IF N = 0 THEN PRINT "";
3150 IF N = 0 THEN PRINT "";
3160 IF N = 0 THEN PRINT "";
3170 IF N = 0 THEN PRINT "";
3180 IF N = 0 THEN PRINT "";
3190 IF N = 0 THEN PRINT "";
3200 IF N = 0 THEN PRINT "";
3210 IF N = 0 THEN PRINT "";
3220 IF N = 0 THEN PRINT "";
3230 IF N = 0 THEN PRINT "";
3240 IF N = 0 THEN PRINT "";
3250 IF N = 0 THEN PRINT "";
3260 IF N = 0 THEN PRINT "";
3270 IF N = 0 THEN PRINT "";
3280 IF N = 0 THEN PRINT "";
3290 IF N = 0 THEN PRINT "";
3300 IF N = 0 THEN PRINT "";
3310 IF N = 0 THEN PRINT "";
3320 IF N = 0 THEN PRINT "";
3330 IF N = 0 THEN PRINT "";
3340 IF N = 0 THEN PRINT "";
3350 IF N = 0 THEN PRINT "";
3360 IF N = 0 THEN PRINT "";
3370 IF N = 0 THEN PRINT "";
3380 IF N = 0 THEN PRINT "";
3390 IF N = 0 THEN PRINT "";
3400 IF N = 0 THEN PRINT "";
3410 IF N = 0 THEN PRINT "";
3420 IF N = 0 THEN PRINT "";
3430 IF N = 0 THEN PRINT "";
3440 IF N = 0 THEN PRINT "";
3450 IF N = 0 THEN PRINT "";
3460 IF N = 0 THEN PRINT "";
3470 IF N = 0 THEN PRINT "";
3480 IF N = 0 THEN PRINT "";
3490 IF N = 0 THEN PRINT "";
3500 IF N = 0 THEN PRINT "";
3510 IF N = 0 THEN PRINT "";
****** Listing of Program 'HM2' ******

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 count, then we read ANOTHER data point and check it all again.
1480 GOTO 1350
1490 REM
1500 ENDO

1510 REM the following subroutine clears the screen
1520 FOR y = 1 TO 24
1530 PRINT
1540 NEXT Y
1550 RETURN
10 SBSUB T20
20 PRINT"For your lesson 3 homework assignment, write a program:"  
30 PRINT"that requests three numbers from the user, and prints them"  
40 PRINT"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 9, be sure your printer is ready:"  
110 PRINT
120 PRINT"Which do you want to do?"  
130 PRINT
140 PRINT"RETURN TO MENU"  
150 PRINT"LIST PROGRAM TO PRINTER"  
160 PRINT"RUN THE PROGRAM"  
170 PRINT
180 INPUT"Enter your choice (1-3):"  
190 IF I$ = "1" THEN RUN "MENU"  
200 IF I$ = "2" THEN PRINT:PRINT"Be sure to type in RUN when the printer is done.":LIST 20:END  
210 IF I$ = "3" THEN SBSUB T20:PRINT"You must type in RUN when the program"  
220 PRINT:PRINT:INPUT"press ENTER to start:"$SBSUB T20:RUN 220  
230 DATA 120  
240 REM This program requests three numbers from the user and puts the  
250 REM numbers into a three element array. It then prints the array to screen  
260 REM so that the numbers appear in ascending order.  
270 REM  
280 REM First, we DIMENSION the array (see lesson 3 part 2):  
290 REM  
300 DIM A(3)  
310 REM  
320 REM Then we use a FOR NEXT loop to ask for three numbers (this is only  
330 REM b. w. of doing this problem, you may want to do it different...).  
340 REM (see lesson 3 part 1 for info on FOR NEXT)  
350 REM  
360 FOR K = 1 TO 3  
370 INPUT A(K)  
380 NEXT K  
390 REM  
400 REM Now we print a general heading so that our output looks a little  
410 REM better:  
420 REM  
430 PRINT  
440 PRINT"These are your numbers in ascending order:"  
450 PRINT  
460 REM
450 REM We can sort the numbers in many ways, but the best straight
460 REM form 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 2 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 T = 1 TO 24
730 PRINT
740 NEXT T
750 RETURN
910 GOSUB 310
920 PRINT "For your lesson 4 homework: improve the program you made in"
930 PRINT "lesson 3. Write the array to disk. Write another small program"
940 PRINT "to read the data back from the disk. THEN print the data in"
950 PRINT "DESCENDING order."
960 PRINT
970 PRINT "I have made a sample program that satisfies the requirements. You"
980 PRINT "may run it or list it to the printer to see how it works."
990 PRINT "If you choose choice B, be sure your printer is ready."
1000 PRINT
1010 PRINT "Which do you want to do?"
1020 PRINT
1030 PRINT "A RETURN TO MENU"
1040 PRINT "B LIST PROGRAM TO PRINTER"
1050 PRINT "C RUN THE PROGRAM"
1060 PRINT
1070 INPUT "Enter your choice": I$
1080 IF I$ = "A" THEN RUN "MENU"
1090 IF I$ = "B" THEN PRINT; PRINT "Be sure to type in RUN when the printer is done.":$LIST 122-79$
1100 IF I$ = "C" THEN GOSUB 810; PRINT; PRINT "You must type in RUN when the program ends.":$PRINT; INPUT; press ENTER to start:"$GOSUB 810; RUN 220
1110 GOTO 110
1120 REM This program requests three numbers from the user and puts the
1130 REM numbers in a three element array. It then prints the array to disk.
1140 REM The second part of the program reads the data back from disk and
1150 REM prints it in DESCENDING order.
1160 REM
1170 REM First, we DIMension the array (see lesson 3 part 2)
1180 REM
1190 DIM A(3)
1200 REM
1210 REM Then we open a file to print the data to disk (see lesson 4 part 1)
1220 REM
1230 OPEN "G:\"; I; "TEST"
1240 REM
1250 REM Then we use a FOR NEXT loop to ask for three numbers (this is only)
1260 REM an example of doing this problem. You may want to do it differently.
1270 REM see lesson 3 part 1 for info on FOR NEXT
1280 REM
1290 REM I chose to write the array to disk as it was entered by the user
1300 REM
1310 FOR I = 1 TO 3
1320 INPUT "Enter a number": I$
1330 PRINT I$
1340 NEXT /
REM Now we close the file (see lesson 4 part 2).
CLOSE
REM We'll stop the program here and let the user know what happened.
PRINT
PRINT"The data is on disk. When you press ENTER the data will be read back from disk and printed in order. Press ENTER to continue.";
PRINT
REM Now we print a general heading so that our output looks a little better.
PRINT
PRINT"These are your numbers in ASCENDING order!";
PRINT
PRINT
PRINT Open the file for input and read the three data points into array A.
OPEN"1",1,"TEST"
FOR I = 1 TO 3
INPUT#1,A(I)
NEXT I
CLOSE
REM We can sort the numbers in many ways, but the most straightforward forward is to use IF statements (see lesson 1 part 21).
REM For this problem, it helps to figure out how many ways three numbers can occur. The answer is 6 - 123,132,123, 312,123,132.
REM That means you can do it with six IF statements.

IF A(1) = A(2) AND A(1) = A(3) THEN PRINT A(1)A(2)A(3)
IF A(1) = A(2) AND A(2) = A(3) THEN PRINT A(2)A(1)A(3)
IF A(1) = A(2) AND A(3) = A(1) THEN PRINT A(1)A(3)A(2)
IF A(1) = A(2) AND A(3) = A(2) THEN PRINT A(2)A(3)A(1)
IF A(1) = A(2) AND A(3) = A(1) THEN PRINT A(3)A(1)A(2)
IF A(1) = A(3) AND A(3) = A(1) THEN PRINT A(1)A(3)A(2)
END
REM
REM The following subroutine clears the screen.
FOR I = 1 TO 24
PRINT
NEXT I
SET_PX
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 stock numbers of furniture that are in your office. (only use a few numbers to start, until you get it debugged!)*  
70 PRINT  
80 PRINT *  
90 PRINT  
100 PRINT * Next, write a program that will let you add a record onto the end of the sequential file of stock numbers.*  
110 PRINT  
120 PRINT  
130 PRINT * Finally, write a program that will search through the disk file for a specific STOCK CLASS of numbers and print them out to screen. (a stock class is the first 4 digits of the stock number)*  
140 PRINT  
150 PRINT *  
160 PRINT *  
170 PRINT  
180 INPUT * press ENTER*; T0  
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 always changes it to scientific notation (see your manual). To prevent that, you must declare the number that you write and read from disk as double precision. The command for that is: DEFDBL (var)*  
240 PRINT  
250 PRINT *  
260 PRINT *  
270 PRINT  
280 PRINT *  
290 PRINT *  
300 PRINT *  
310 PRINT *  
320 PRINT  
330 PRINT *  
340 PRINT  
350 PRINT * As you see in the example, I was not changed to scientific notation*  
360 PRINT * when it was declared double precision.*  
370 PRINT  
380 INPUT * press ENTER*; T0  
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 sequential file. Here is a summary of that example.*  
430 PRINT  
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 ID 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" RETURN TO MENU*
840 PRINT" 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":LL:11;11;"09-1140
890 GOTO 810 
900 REM
910 REM This is the first program. It requests stock #s and writes them to disk 
920 REM
**** Listing of Program 'HWS' ****

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 I, 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 I, X
1340 PRINT2, X
1350 GOTO 1320
1360 REM
1370 REM When the old file is empty, the program jumps to the CLOSEI statement
1380 REM
**Listing of Program 'HN5'**

```
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 FS
1540  END
1550  REM
1560  REM This is the third program. It searches a disk file and tells you
1570  REM the number of occurrences of a stock class
1580  REM
1590  REM First, declare our variable as double precision and open a file
1600  REM
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', I, 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(I) THEN GOTO 1820
1740  INPUT # I
1750  IF N = I THEN PRINT I
1760  IF N = 1 THEN GOTO 1730
1770  IF INT(1E-000000000001) = N THEN PRINT I
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$
```

492
**** Listing of Program 'HMS' ****

1850 IF TS = "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"",1,F$
1990 N = 0  : "N IS A COUNTER FOR THE ARRAY
2000 IF EOF(1) THEN GOTO 2040
2010 N = N+1
2020 INPUT1,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"",1,F$
2160 FOR Y = 1 TO N-1
2170 PRINT$I,A(Y)
2180 NEXT Y
2190 CLOSE
2200 END
2210 REM
2220 REM 'DILA' WE DID ALL OUR EDITING WITH THE SAME FILE'
2230 REM
2240 REM
2250 REM this subroutine clears the screen
2260 REM
2270 FOR I = 1 TO 24
2280 PRINT
2290 NEXT I
2300 RETURN
APPENDIX E

REPORT PROGRAM LISTING
***** Listing of Program REPORT *****

10 REM This program started on 29 June 1983
20 REM Version Number is 1 August 1983
30 REM
40 REM Author - Cast 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 - S9 = arrays that hold scores for tests 1-9
120 REM 1.J.A.L.R.T.T.J-T = temporary variables & counters
130 REM
140 CLEAR 10000
150 DIM N$(60), S1$(60), S2$(60), S3$(60), S4$(60), S5$(60), S6$(60), S7$(60), S8$(60), S9$(60)
160 ON ERROR GOTO 1880
170 x=0: t=1
180 REM
190 REM WE READ IN ALL THE SCORES WITH THE FOLLOWING
200 REM ROUTINES
210 REM
220 GOSUB 1740
230 PRINT: PRINT: PRINT: PRINT:
240 PRINT: "COMPUTER ASSISTED INSTRUCTION"
250 PRINT: "IN BASIC"
260 PRINT:
270 PRINT:PRINT:PRINT
280 PRINT: "THIS PROGRAM READS IN ALL THE SCORES FROM THE DIFFERENT TEST"
290 PRINT: "FILES."
300 PRINT:
310 PRINT: "IF YOU WISH TO CONTINUE WITH IT, BE SURE THE SCORES YOU WANT"
320 PRINT: "PRINTED OUT ARE ON ONE OF YOUR ACTIVE DRIVES."
330 PRINT:
340 INPUT "DO YOU WISH TO CONTINUE? (Y/N)": N$:
350 IF N$ = "Y" THEN GOSUB 1740: PRINT: "GOING TO MENU: (Y/N)MENU"
360 GOSUB 1740
370 PRINT: "PROGRAM STARTING NOW"
380 PRINT:
390 INPUT "FILE NAME:": "$I. SCORE"
400 IF Y = "T" THEN GOTO 440
410 IF EOF(1) THEN GOTO 440
420 INPUT: N$(4): M$(4)
430 GOTO 440
440 CLOSE: =G2: OPEN "$I. SCORE"
450 IF Y = "T" THEN GOTO 440
460 IF EOF (1) THEN GOTO 440
****** Listing of Program 'REPORT' ******

470 #1=1
480 INPUT #1, N$ #1, #2(N)
490 GOTO 470
500 CLOSE=#1;OPEN#1..."SCORE#"
510 IF Y = 7 THEN GOTO 520
520 IF EOF(#1) THEN GOTO 530
530 N=#1
540 INPUT #1, N$ #1, #3(X)
550 GOTO 520
560 CLOSE=#1;OPEN#1..."SCORE#"
570 IF Y = 7 THEN GOTO 530
580 IF EOF(#1) THEN GOTO 530
590 Y = Y+1
600 INPUT #1, N$ #1, #5(X)
610 GOTO 520
620 CLOSE=#1;OPEN#1..."SCORE#"
630 IF Y = 7 THEN GOTO 530
640 IF EOF(#1) THEN GOTO 530
650 X = X+1
660 INPUT #1, N$ #1, #7(X)
670 GOTO 520
680 CLOSE=#1;OPEN#1..."SCORE#"
690 IF Y = 7 THEN GOTO 530
700 IF EOF(#1) THEN GOTO 700
710 Y = Y+1
720 INPUT #1, N$ #1, #9(X)
730 GOTO 700
740 CLOSE=#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 = 1 TO X
810 IF Y$#N = CHRS(128) THEN GOTO 880: If leg already checked, then skip it
820 IF Y$#N = Y$#Y THEN GOSUB 150: If name is duplicate, call subroutine
830 NEXT N
840 PRINT ":
850 Y = Y+1
860 IF - - X = 1 THEN GOTO 310
870 REM
880 REM WE CAN SORT THEM HERE, BUT IT TAKES A WHILE SO
890 REM USES HAS THE OPTION TO GO ON WITHOUT A SORT.
900 REM
910 PRINT "The computer can sort out the names if you wish, however."
920 PRINT
930 REM
940 REM
**** Listing of Program "REPORT" ****

970 PRINT: BASIC sorts take a few moments. **PRINT: PRINT: INPUT"Do you want to sort the names
**PRINT: INPUT"**s
980 IF LEFT$(s,1) = "Y" OR LEFT$(s,1) = "y" THEN GOSUB 1310
990 REM
1000 REM THIS PRINTS OUT REPORT TO SCREEN
1010 REM
1020 PRINT
1030 LINE INPUT What is today's date (month/day/year)? "*"$
1040 PRINT "*"$; "right (Y/N):" INPUT T$
1050 IF LEFT$(T,1) = "N" THEN GOTO 990
1060 IF LEFT$(T,1) = "y" THEN GOTO 990
1070 PRINT
1080 INPUT "do you want scores sent to line printer (Y/N): "$
1090 IF LEFT$(T,1) = "Y" THEN GOTO 1220
1100 REM GOSUB 170
1110 PRINT: REPORT: DATE: "*"
1120 PRINT: PRINT
1130 PRINT TAB(15): "NAME"$; TAB(20): "TEST 1"$; TAB(18): "TEST 2"$; TAB(20): "TEST 3"$; TAB(18): "TEST 4"$; TAB(16): "TEST
"$
1140 PRINT: PRINT: PRINT
1150 PRINT: PRINT
1160 PRINT: PRINT: FOR X = 1 TO Y+1
1170 IF W$(X) = CHR$(120) THEN GOTO 1220
1180 IF LEN(W$(X)) = 0 THEN GOTO 1220
1190 PRINT W$(X);$
1200 PRINT TAB(15): IF S$(1) = 0 THEN PRINT: ***: ELSE PRINT: $1$(X);$
1210 PRINT TAB(23): IF S$(1) = 0 THEN PRINT: ***: ELSE PRINT: $2$(X);$
1220 PRINT TAB(31): IF S$(1) = 0 THEN PRINT: ***: ELSE PRINT: $3$(X);$
1230 PRINT TAB(43): IF S$(1) = 0 THEN PRINT: ***: ELSE PRINT: $4$(X);$
1240 PRINT TAB(54): IF S$(1) = 0 THEN PRINT: ***: ELSE PRINT: $5$(X);$
1250 PRINT TAB(66): IF S$(1) = 0 THEN PRINT: ***: ELSE PRINT: $6$(X);$
34A FOR X = 1 TO 24
350 IF N@X) = CHR$(124) THEN GOTO 1450
3550 IF LEN(N@X)) = 0 THEN GOTO 1450
360 LPRINT N@X:
365 LPRINT TAB(20):IF S1(X) = 0 THEN LPRINT "***";ELSE LPRINT S1(X)
370 LPRINT TAB(20):IF S2(X) = 0 THEN LPRINT "***";ELSE LPRINT S2(X)
375 LPRINT TAB(20):IF S3(X) = 0 THEN LPRINT "***";ELSE LPRINT S3(X)
380 LPRINT TAB(20):IF S4(X) = 0 THEN LPRINT "***";ELSE LPRINT S4(X)
385 LPRINT TAB(20):IF S5(X) = 0 THEN LPRINT "***";ELSE LPRINT S5(X)
390 LPRINT TAB(20):IF S6(X) = 0 THEN LPRINT "***";ELSE LPRINT S6(X)
395 LPRINT 1450 NEXT X
400 REM
405 REM FOLLOWING LINE CAUSES FORMFEED ON MOST PRINTERS
410 REM YOU MAY HAVE TO CHANGE IT FOR YOUR MACHINE
415 REM
420 LPRINT CHR$(140)
425 LPRINT END OF PROGRAM - PRESS BREAK OR CONTROL C TO END
430 GOTO 1520
435 REM
440 REM THIS ROUTINE CONSOLIDATES DUPLICATE NAMES
445 REM
450 IF S1(X) = S1(Y) THEN S1(X) = S1(Y)
455 IF S2(X) = S2(Y) THEN S2(Y) = S2(X)
460 IF S3(X) = S3(Y) THEN S3(Y) = S3(X)
465 IF S4(X) = S4(Y) THEN S4(Y) = S4(X)
470 IF S5(X) = S5(Y) THEN S5(Y) = S5(X)
475 IF S6(X) = S6(Y) THEN S6(Y) = S6(X)
480 N@X) = CHR$(124):"replace the duplicate name with a flag"
485 RETURN
490 REM
495 REM THIS IS ERROR TRAP - IT PREVENTS PROGRAM STOPPING
500 REM WHEN ONE OF THE SCORE FILES IS NOT FOUND
505 REM
510 PRINT "FILE";X: "NOT FOUND"
515 R=7
520 RESUME NEXT
525 REM
530 REM THIS CLEARS THE SCREEN ON MOST MACHINES
535 REM
540 FOR X = 1 TO 24
545 PRINT
550 NEXT X
555 RETURN
560 REM
565 REM this is the bubble sort used to put the names in order
LISTING OF PROGRAM 'REPORT' ****

1800 REM
1810 I = X - 1
1820 PRINT
1830 PRINT "Sorting through the names now:"
1840 FOR J = Y TO Z STEP -1
1850 Y = J + 1
1860 FOR L = X TO K STEP -1
1870 IF N(J) > N(L) THEN GOTO 1990
1880 REM
1890 REM save first value
1900 REM
1910 T = N(J): 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(J) = N(J): S1(L) = S1(J): S2(L) = S2(J): S3(L) = S3(J)
1970 NEXT L
1980 NEXT J
1990 PRINT "":
2000 NEXT J
2010 PRINT
2120 RETURN
SELECTED BIBLIOGRAPHY
REFERENCES CITED


18. Strategic Air Command Regulation 300-4

19. Tactical Air Command Regulation 300-12
