COMPUTER ASSISTED INSTRUCTION
IN BASIC

Captain Danny J. Creagan, USAF

LESSR 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.
**REPORT DOCUMENTATION PAGE**

<table>
<thead>
<tr>
<th>1. REPORT NUMBER</th>
<th>2. GOVT ACCESSION NO.</th>
<th>3. RECIPIENT'S CATALOG NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSSR 29-83</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE (and Subtitle)</th>
<th>5. TYPE OF REPORT &amp; PERIOD COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER ASSISTED INSTRUCTION IN BASIC</td>
<td>Master's Thesis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. AUTHOR(s)</th>
<th>8. CONTRACT OR GRANT NUMBER(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danny J. Creagan, Captain, USAF</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. PERFORMING ORGANIZATION NAME AND ADDRESS</th>
<th>10. PROGRAM ELEMENT PROJECT, TASK AREA &amp; WORK UNIT NUMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>School of Systems and Logistics</td>
<td></td>
</tr>
<tr>
<td>Air Force Institute of Technology, WPAFB OH</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. CONTROLLING OFFICE NAME AND ADDRESS</th>
<th>12. REPORT DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department of Communication</td>
<td>28 September 1983</td>
</tr>
<tr>
<td>AFIT/LSH, WPAFB OH 45433</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. NUMBER OF PAGES</th>
<th>14. MONITORING AGENCY NAME &amp; ADDRESS (IF different from Controlling Office)</th>
<th>15. SECURITY CLASS. (OF THIS REPORT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>510</td>
<td></td>
<td>UNCLASSIFIED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. DISTRIBUTION STATEMENT (OF THIS REPORT)</th>
<th>17. DISTRIBUTION STATEMENT (OF THE ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release; distribution unlimited</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. SUPPLEMENTARY NOTES</th>
<th>19. KEY WORDS (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release: LAW AFH 150-17,</td>
<td></td>
</tr>
<tr>
<td>EYTH Z. WOLVERTON</td>
<td>Computers, Programmed Instruction, Computer Aided Instruction, Programming Languages,</td>
</tr>
<tr>
<td>Wright-Patterson AFH OH 45433</td>
<td>Microcomputers, Thesis Chairman: Jeffrey C. Daneman, GS-13, AFIT/LSQ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20. ABSTRACT (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thesis Chairman: Jeffrey C. Daneman, GS-13, AFIT/LSQ</td>
</tr>
</tbody>
</table>

**DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE**
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
TABLE OF CONTENTS

CHAPTER

1. BACKGROUND ........................................ 1

   COMPUTER ASSISTED INSTRUCTION IN BASIC .......... 1

   ISSUE ............................................ 1

   PROBLEM ......................................... 2

   RESEARCH QUESTION ............................... 6

2. METHODOLOGY ...................................... 8

   THE CAI TECHNIQUES ON A MICROCOMPUTER ......... 8

   Physical Constraints on Design of CAI ........... 11

   Development of the CAI Draft .................... 13

   Coding of the CAI Program ....................... 14

   Validation ...................................... 14

3. CAI PROGRAM FORMAT .............................. 16

   PROGRAM LIBRARY ................................ 16

   Program Purpose ................................ 16

   Flowcharts .................................... 17

   Lessons ....................................... 17

   Flowchart for Main Menu ......................... 19

   Flowchart for Lessons One Through Six .......... 20
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<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 OAR (Data Automation Requirement) by the unit (the OAR, in this case, was a formal request which outlined a specific purpose for the computer), and approval of the OAR by the appropriate MAJCOM. Software support was limited to off-the-shelf packages. All of the computers were purchased with Beginners All Purpose Instruction Code (BASIC), a word processor and a data base management program. After purchase, the computers were considered stand alone systems and no increase in manpower to support them was authorized (TACR 300-12, SAC 300-4).

Since the decision to purchase the units, approximately 400 have been installed at a cost of $1000 to 20,000 dollars each. The cost was determined by number of peripherals (printers, plotters, disk drives, etc.) that were requested with computers (Coward, 82). The expected workload for the computers is light and the machines will be available for other mission-related tasks if the users are able to create some of their own software (Lyon, 83).

PROBLEM

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

The BASIC programming language is the best choice for training new operators how to program because BASIC is available on virtually every microcomputer (the Cromemcos purchased by the Air Force included a dialect of BASIC called Microsoft BASIC). Another reason for BASIC is that many good, mission-related programs have already been written in it, showing the language's capability to quickly do time consuming tasks (Carter, Nunley, 81). Additionally, the popularity, usefulness, and ease of modification of BASIC programs has caused the senior members of the Air Force to approve BASIC as an accepted programming language (HQ USAF Message, 82). These reasons have led to a
consensus by HQ TAC personnel that a CAI (Computer Assisted Instruction) program in the BASIC language would provide the most effective solution for the microcomputer training problem. The best dialect of BASIC to use would be the Microsoft BASIC dialect because it is available on the majority of microcomputers (Burke, 82; Blackwood, 82; Lewis, 82; Welch and others, 82).

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

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

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

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

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

The second step would be the development of the "frames" for the program. A CAI frame is a portion of a CAI program that contains a single learning event. The events are typically broken into information, question, student response, and feedback/answer sentences (Meredith, 71; Freedman, 81). A microcomputer usually has limited memory and storage space. The Cromemco machines, when BASIC is implemented, have around 33 kilobytes (33k) of memory. In
addition, the off line storage of microcomputers is sometimes limited to one or two small floppy disk drives (Coward, 82). Therefore, an exhaustive instruction in the language would not be possible, because there would not be enough room for the program in the computer. The frames would have to be isolated into main topics which would sufficiently instruct the student in the language, yet still fit within the storage space and memory limitations of a one disk, 33k machine.

The third step would involve the coding and debugging of the CAI program itself. While the frames should be simple to code, there will have to be hundreds of them to adequately cover the subject. Such a large program will need careful debugging to eliminate errors. Consultation with non-programmers to ensure "user friendliness", and using their feedback to fine tune the CAI, would conclude the research and development of the CAI program.
CHAPTER 2

METHODOLOGY

THE CAI TECHNIQUES ON A MICROCOMPUTER

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

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

The Air Force has been interested in how to use CAI
effectively since the early 1970s. They formed an office at Keesler AFB in Mississippi (the 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 Z80 CPU card - see the Apple operating manuals for further explanation of how to run Microsoft BASIC on that equipment). This will enables the package to run on the microcomputers which make up the majority of microcomputer installations (Cokerly, 83; Coward, 82; Greene, 82).

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

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

Development of the CAI Draft

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

Questioning techniques (test items). Almost any type of test item can be used in CAI. The limiting factors are the amount of memory required to store the possible test answers (Holtzman and Silvern, 70). It would be obvious from this, that essay answers would be very demanding of memory,
and multiple choice, matching, true-false, or one word responses would be more memory conservative (Burke, 82). Because of the equipment limitations discussed earlier, I have not used essay questions.

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

**Coding of the CAI Program**

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

**Validation**

The CAI program was validated at several stages during its development. As each block was completed, it was tested at AFIT. The purpose of the initial testing was to
eliminate logic errors in the CAI program itself. Therefore, the target audience during the debug phase was knowledgeable in what makes a good program. The program will be sent to the focal point offices and released for testing within the commands. Feedback from these offices will be used to revise the programs as needed.
CHAPTER 3

CAI PROGRAM FORMAT

PROGRAM LIBRARY

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

Program Purpose

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

Flowcharts

Lessons

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

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

ON CHOICE:
RUN PROGRAM SELECTION
Flowcharts for Lessons One Through Six

Lesson Section Subroutine

MENU CHOICES
1. Take Lesson in Fixed Order
2. Take Lesson According to Student Wishes
3. Goto Next Part
4. Run Main Menu

CALL A SECTION

MENU CHOICES
1. Review a Section
2. Run Main Menu

DISPLAY MENU

INPUT

GOTO NEXT PART

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)
Report Program

The report program reads in all the scores, assigns them to the proper students, and prints out a summary to the monitor or to the line printer. If a student hasn't completed a test for one of the lessons, a set of asterisks appears for the test score. The user of this program has the option of receiving the report with student names in alphabetical order. A sample report is on the following page.
<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></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>39. OPEN</td>
<td>4/1</td>
</tr>
<tr>
<td>40. PRINT</td>
<td>1/1</td>
</tr>
<tr>
<td>41. PRINT#</td>
<td>4/2</td>
</tr>
<tr>
<td>42. READ</td>
<td>2/2</td>
</tr>
<tr>
<td>43. REMark</td>
<td>2/1</td>
</tr>
<tr>
<td>44. RESTORE</td>
<td>2/2</td>
</tr>
<tr>
<td>45. RETURN</td>
<td>5/1</td>
</tr>
<tr>
<td>46. RUN</td>
<td>2/1</td>
</tr>
<tr>
<td>47. SAVE</td>
<td>2/1</td>
</tr>
<tr>
<td>48. Sequential Files</td>
<td>4/1</td>
</tr>
<tr>
<td>49. Software</td>
<td>1/1</td>
</tr>
<tr>
<td>50. Statements and Programs</td>
<td>1/1</td>
</tr>
<tr>
<td>51. String Arrays</td>
<td>6/1</td>
</tr>
<tr>
<td>52. String Assignment</td>
<td>6/1</td>
</tr>
<tr>
<td>53. String Functions</td>
<td>6/1</td>
</tr>
<tr>
<td>54. String Variables (intro)</td>
<td>1/2</td>
</tr>
<tr>
<td>55. Subroutines</td>
<td>5/1</td>
</tr>
<tr>
<td>56. User Defined Functions</td>
<td>5/2</td>
</tr>
<tr>
<td>57. Using Arithmetic in Computers</td>
<td>1/2</td>
</tr>
<tr>
<td>58. Variables (general)</td>
<td>1/2</td>
</tr>
<tr>
<td>59. nC(hange)</td>
<td>6/2</td>
</tr>
<tr>
<td>60. nD(lete)</td>
<td>6/2</td>
</tr>
<tr>
<td>61. nS(earch)</td>
<td>6/2</td>
</tr>
<tr>
<td>62. nSPACEBAR</td>
<td>6/2</td>
</tr>
<tr>
<td>63. X(tend)</td>
<td>6/2</td>
</tr>
</tbody>
</table>

28
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/Rand Access</td>
<td>FIELD, GET, PUT, LOF OPEN&quot;R&quot;,1,&quot;lfn&quot;</td>
</tr>
<tr>
<td></td>
<td>LSET, RSET</td>
</tr>
<tr>
<td>3. Graphics</td>
<td>POINT, SET, RESET (TRS-80)</td>
</tr>
<tr>
<td>4. ASCII code</td>
<td>SAVE &quot;lfn&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 GOSUB, RESUME,</td>
</tr>
<tr>
<td></td>
<td>ERR, ERL</td>
</tr>
<tr>
<td>7. Defining precision</td>
<td>DEFDBL, DEFSNG, DEFINT, CVI, CVSD,</td>
</tr>
<tr>
<td></td>
<td>CDBL, 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),
RENUN (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 .............. VARPTA

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.
APPENDICES
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, 1933, 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>Section</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>
**** Listing of Program 'MENU' ****

10 REM ** THIS PROGRAM STARTED ON 27 MARCH 1983
15 REM ** MODIFIED FOR LDOS DOUBLE SIDED DRIVE ON 20 JUNE 1983
20 REM ** AUTHOR - CAPT D. CREAGAN - AFIT
30 REM ** TITLE - MENU PROGRAM
40 REM **
50 REM ** SUBROUTINE(S):
60 REM **
70 REM ** GOSUB 21000 = ROUTINE TO CLEAR SCREEN
80 REM ** THE CLEAR SCREEN FUNCTION
90 REM ** IS NON-STANDARD. THIS
100 REM ** SUBROUTINE PRINTS 24 LINE
110 REM ** FEEDS TO INSURE SCREEN IS
120 REM ** CLEARED ON MOST TERMINALS
130 REM **
140 GOSUB 520
150 PRINT* COMPUTER ASSISTED INSTRUCTION IN BASIC*
160 PRINT
170 PRINT* by: Captain Dan Creagan*
180 PRINT* Air Force Institute of Technology*
190 PRINT
200 PRINT "This is the menu for computer assisted instruction."
210 PRINT "It is meant to be used with a BASIC manual or with the program above."
220 PRINT "It is meant to be used with a BASIC manual or with the program above."
230 PRINT
240 PRINT
250 PRINT "If you wish to give me feedback, or get information about the program, please contact me at 512-413-7210."
260 PRINT "If you wish to give me feedback, or get information about the program, please contact me at 512-413-7210."
270 PRINT "If you wish to give me feedback, or get information about the program, please contact me at 512-413-7210."
280 PRINT
290 INPUT "Press ENTER to continue":
300 GOSUB 520
310 PRINT "MENU CHOICE":
320 PRINT
330 PRINT "Select the program you wish to run from the list below, and press the number that is next to your selection."
340 PRINT "Select the program you wish to run from the list below, and press the number that is next to your selection."
350 PRINT "Select the program you wish to run from the list below, and press the number that is next to your selection."
360 PRINT
370 PRINT 1. LESSON ONE
380 PRINT 2. TEST ONE
390 PRINT 3. LESSON TWO
400 PRINT 4. TEST TWO
410 PRINT 5. LESSON THREE
420 PRINT 6. TEST THREE
430 PRINT
440 INPUT "Which number do you want":
450 ON T GOTO 450,452,454,456,460,462,464,466,470,472,474,480,490
### Listing of Program 'MENU' ###

07/10/83 - 00:59:51

450 RUN"LESSON1"
452 RUN"TEST1"
454 RUN"LESSON2"
456 RUN"TEST2"
460 RUN"LESSON3"
462 RUN"TEST3"
464 RUN"LESSON4"
466 RUN"TEST4"
470 RUN"LESSON5"
472 RUN"TEST5"
474 RUN"LESSON6"
480 RUN "TEST6"
490 END

500 REM **
510 REM ** FOLLOWING SUBROUTINE IS USED TO CLEAR SCREEN
520 FOR I = 1 TO 24
530 PRINT " "
540 NEXT I
550 RETURN
**** Listing of Program 'LESSON1' ****

1000 REM ** THIS PROGRAM STARTED ON 07 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: I 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 REQUIRED: BASIC reference manual'
1190 PRINT
1200 PRINT
1210 PRINT
1220 PRINT 'press the ENTER key to continue; TS$
1230 GOSUB 9970
1240 GOSUB 10040
1250 PRINT 'I'm taking this part in its entirety; TS$
1260 PRINT 'I wish to review selected areas; TS$
1270 PRINT 'I want to go to the second part; TS$
1280 PRINT 'I want to return to the Menu; TS$
1290 PRINT
1300 PRINT 'Press either capital A, B, C, or D and then press ENTER; TS$
1310 IF TS$ = 'D' GOTO 10160
1320 IF TS$ = 'C' GOTO 10170
1330 IF TS$ = 'B' GOTO 1450
1340 IF TS$ = 'A' GOTO 1250
1350 GOSUB 1560
1360 GOSUB 1920
1370 GOSUB 2790
1380 GOSUB 7750
1390 GOSUB 4800
1400 GOSUB 5950
1410 GOSUB 7820
1420 GOSUB 8500
1430 PRINT 'GOING TO SECOND PART - PLEASE STAND BY'
1440 GOTO 10170
1450 GOSUB 9970
Listing of Program 'LESSON' 07/10/83 - 01:02:41

1460 GOSUB 10040
1470 PRINT
1480 PRINT"Please type in the number beside the area you wish"
1490 PRINT"to review (1 through 8) and then press ENTER - press 9 and"
1500 PRINT"press ENTER to return to the Menu."
1510 PRINT
1520 INPUT"What is your choice":N
1530 IF N = 0 GOTO 10160
1540 GOSUB 1560 .1920 .2790 .3750 .4800 .5950 .7820 .3900
1550 GOTO 1450
1560 GOSUB 9970
1570 PRINT"Introduction"
1580 PRINT
1590 PRINT"Throughout all your lessons, you should have your BASIC manual"
1600 PRINT"handy. If you find yourself stumped by a question, you should"
1610 PRINT"look up the answer in the book. If you can't find it after"
1620 PRINT"an honest attempt, then make a guess and then go on. You will"
1630 PRINT"have an opportunity to review each section again."
1640 PRINT
1650 PRINT"Beginning with this lesson, you will have homework assigned"
1660 PRINT"at the end of each test. If you do the homework, you will"
1670 PRINT"learn more, and, with the techniques you learn, you will find"
1680 PRINT"that you can tackle small programming jobs as soon as you"
1690 PRINT"complete the course."
1700 PRINT
1710 INPUT"press ENTER":T:GOSUB 9970
1720 PRINT
1730 PRINT"Introduction"
1740 PRINT
1750 PRINT"Throughout the next six lessons you will be learning"
1760 PRINT"about computers and what they do. Although the course is"
1770 PRINT"titled 'Computer Assisted Instruction in BASIC', you will"
1780 PRINT"also need to learn the terminology of computers, not just"
1790 PRINT"the BASIC programming language. This first lesson will"
1800 PRINT"start with some fundamental ideas, and expand them as"
1810 PRINT"we go along."
1820 PRINT
1830 PRINT"We use computers to process DATA and give us answers to our"
1840 PRINT"problems. To process this DATA, we must communicate with the"
1850 PRINT"computer using two basic computer components. Those components are called: HARDWARE AND SOFTWARE."
1860 PRIN
1870 INPUT"press ENTER":T
1880 GOSUB 9970
1890 IF T = 9 GOTO 1560
1900 RETURN
Listing of Program 'LESSON1' 07/10/87 - 01:03:41

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 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 discs."
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 PRINT
2380 INPUT "Press ENTER to continue":$1

45
2290 GOSUB 9970
2300 PRINT "Here is a little quiz - answer in capital letters"
2400 PRINT "and do not include extra spaces or words"
2410 PRINT
2420 PRINT "What component (HARDWARE or SOFTWARE) is mainly"
2430 PRINT "used to transform your inputs into a form the CPU can"
2440 INPUT "understand"; T$  
2450 PRINT
2460 IF T$ = "SOFTWARE" THEN PRINT "WRONG - the correct answer is SOFTWARE"
2470 IF T$ = "SOFTWARE" THEN PRINT "CORRECT" 
2480 PRINT
2490 INPUT "press ENTER to continue"; T$ 
2500 GOSUB 9970 
2510 INPUT "What do the letters CPU stand for"; T$ 
2520 PRINT
2530 IF T$ = "CENTRAL PROCESSING UNIT" GOTO 2560 
2540 PRINT "WRONG - the correct answer is CENTRAL PROCESSING UNIT"
2550 GOTO 2570 
2560 PRINT "CORRECT - now we are learning something!" 
2570 PRINT
2580 INPUT "press ENTER to continue"; T$ 
2590 GOSUB 9970 
2600 PRINT "Which of the following can be considered an OUTPUT device?"
2610 PRINT
2620 PRINT "A. Terminals"
2630 PRINT "B. Tapes"
2640 PRINT "C. Disks"
2650 PRINT "D. ALL of the above" 
2660 PRINT
2670 INPUT "which letter do you select"; T$ 
2680 IF LEN(T$) = 1 GOTO 2670 
2690 PRINT
2700 IF T$ = "D" GOTO 2730 
2710 PRINT "WRONG - D. ALL of the above is the correct answer"
2720 GOTO 2750 
2730 PRINT "CORRECT" 
2740 PRINT
2750 INPUT "press ENTER to continue"; T$ 
2760 GOSUB 1010 
2770 IF T$ = "S" GOTO 1920 
2780 RETURN 
2790 GOSUB 9970 
2800 PRINT "SOFTWARE" 
2810 PRINT
2820 PRINT "Software is a collection of written rules that control"
2830 PRINT "the computer. Software can be divided into two"
Listing of Program 'LESSON1'

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":F$'
2940 GOSUB 9970
2950 PRINT"SOFTWARE (cont)"
2960 PRINT
2970 PRINT "The OPERATING SYSTEM supervises the various capabilities of the computer and cannot be altered by the user."  
2980 PRINT "It OVERSEES the operation, and senses when a keystroke"
2990 PRINT "is made, a button is pushed, or a request made."
3000 PRINT
3010 PRINT "One part of the operating system is called the LANGUAGE"
3020 PRINT "PROCESSOR. The LANGUAGE PROCESSOR translates the instructions"
3030 PRINT "of a user-written program into electronic instruc-
3040 PRINT "tions that the computer can understand."
3050 PRINT
3060 PRINT "The rules, or grammar, that you use to write your software"
3070 PRINT "are described by the kind of computer language you use."
3080 PRINT
3090 INPUT "Press ENTER to continue":F$
3100 GOSUB 9970
3110 PRINT"SOFTWARE (cont)"
3120 PRINT
3130 PRINT "Because user-written programs and operating systems are"
3140 PRINT "both designed by humans, it is possible to develop a human-
3150 PRINT "oriented language that both can use. That is, a language"
3160 PRINT "that lets you write programs using easily mastered rules and"
3170 PRINT "conventions that are also understood by the operating"
3180 PRINT "system. Once we get the operating system to understand"
3190 PRINT "the instruction, it can make the computer do its job."
3200 PRINT
3210 PRINT
3220 PRINT
3230 PRINT
3240 PRINT
3250 PRINT
3260 INPUT "Press ENTER to continue":F$
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""
3310 PRINT "spaces or words."
3320 PRINT
3330 INPUT "Press ENTER to continue": T$""
3340 GOSUB 9970
3350 PRINT "is a peripheral, such as a line printer, hardware or software""
3360 PRINT
3370 PRINT "A Hardware"
3380 PRINT "B Software"
3390 PRINT
3400 PRINT "Choose A or B - press the letter and then press ENTER""
3410 PRINT
3420 INPUT "Which letter": T$""
3430 IF LENT$; > 1 GOTO 3400
3440 PRINT
3450 IF T$ = "A" GOTO 3490
3460 PRINT "Wrong - the correct answer is A (Hardware)."
3470 GOTO 3490
3480 PRINT "Correct""
3490 PRINT
3500 INPUT "Press ENTER to continue": T$"
3510 GOSUB 9970"
3520 PRINT "An operating system translates user-written code into a""
3530 PRINT "form that the operator can understand. (TRUE or""
3540 PRINT "FALSE)"
3550 PRINT
3560 PRINT "A True"
3570 PRINT "B False"
3580 PRINT
3590 PRINT "Choose the letter corresponding to the correct answer""
3600 PRINT
3610 INPUT "Which letter (A or B)"": T$"
3620 PRINT
3630 IF T$ = "B" GOTO 3690
3640 PRINT "Incorrect - the right answer is B (False)"
3650 PRINT
3660 PRINT "It is the language processor's job to convert user code""
3670 PRINT "into a form the computer can understand."
3680 GOTO 3700""
3690 PRINT "You are RIGHT"
3700 PRINT
3710 INPUT "Press ENTER to continue": T$"
3720 GOSUB 9920"
3730 IF T$ = "B" GOTO 3790
3740 RETURN
3750 GOSUB 9920"
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.

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.

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.

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 10, 20, 50, 100, and 50. 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.
4120 INPUT"press ENTER for an example of the PRINT instruction";T$  
4130 GOSUB 9970  
4140 PRINT"General Information (cont)"  
4150 PRINT  
4160 PRINT"The PRINT statement works like this: to find the average:"  
4170 PRINT"of five numbers and write the result on your terminal:"  
4180 PRINT"you can use the following BASIC statement:"  
4190 PRINT  
4200 PRINT"PRINT (19+80+2+100+68)/5"  
4210 PRINT  
4220 PRINT"In this example, the BASIC verb PRINT tells the operating:"  
4230 PRINT"to write the instruction following it to the terminal:"  
4240 PRINT"The data, or recipients of the verb PRINT, are the numbers:"  
4250 PRINT"and symbols to the right of the PRINT word. The symbols:"  
4260 PRINT"are used the same way that you use them on a calculator:"  
4270 PRINT  
4280 PRINT  
4290 PRINT"Press ENTER to continue with General Information";T$  
4300 GOSUB 9970  
4310 PRINT"BASIC is used by nearly every micro and mainframe computer."  
4320 PRINT  
4330 PRINT"There are many "dialects" of BASIC and they are not all:"  
4340 PRINT"compatible with each other. For instance, a BASIC program:"  
4350 PRINT"written in Honeywell BASIC or Applesoft BASIC will not:"  
4360 PRINT"run on your computer unless it is modified."  
4370 PRINT  
4380 PRINT"The name for the BASIC on your computer is Microsoft BASIC."  
4390 PRINT"Microsoft BASIC is supported by some microcomputers than:"  
4400 PRINT"any other dialect. It is extremely powerful, and matches:"  
4410 PRINT"the computing capability of most other languages. It:"  
4420 PRINT"is slower than some, but the slowness is relative. Most:"  
4430 PRINT"calculations only take milli-seconds."  
4440 PRINT  
4450 PRINT"Press ENTER to continue"  
4460 GOSUB 9970  
4470 PRINT"Why is BASIC a good general programming language to learn?"  
4480 PRINT  
4490 PRINT"It can be used by most students and programmers:"  
4500 PRINT"you don't need to know a lot about computers to use it:"  
4510 PRINT"it is available on most computers:"  
4520 PRINT"ALL of the above"  
4530 PRINT  
4540 PRINT"Press the letter corresponding to the correct answer:"  
4550 PRINT"Be sure to enter only capital letters"  
4560 END
**Listing of Program 'LESON'**

430 INPUT "What is your selection?" T$  
430 PRINT  
440 IF T$ = "D" GOTO 4740  
450 PRINT "WROG - the correct answer is D 'ALL of the above'"  
460 PRINT  
470 GOTO 4750  
470 PRINT "correct"  
480 PRINT  
490 INPUT "press ENTER to continue"  
490 GOSUB 1210  
500 IF T$ = "N" GOTO 4750  
510 RETURN  
520 GOSUB 9970  
530 PRINT  
540 PRINT "STATEMENTS and PROGRAMS"  
550 PRINT  
560 PRINT "the instructions that we saw in the previous example is a"  
570 PRINT "one line command to the computer. When we combine several"  
580 PRINT "statements, we get a more useful COMPUTER PROGRAM."  
590 PRINT  
600 PRINT  
610 PRINT "the COMPUTER PROGRAM acts as a series of directions for"  
620 PRINT "the machine to follow."  
630 PRINT  
640 PRINT  
650 PRINT "the statements that make up the program are expressed as"  
660 PRINT "BASIC words which denote an action to be taken - THEY"  
670 PRINT "appear SEQUENTIALLY IN NUMERED PROGRAM LINES. Usually"  
680 PRINT "along with the data that is to be acted upon."  
690 PRINT  
700 PRINT  
710 INPUT "press ENTER to continue with STATEMENTS & PROGRAMS"  
720 GOSUB 9970  
730 PRINT  
740 PRINT "Each BASIC statement consists of a specific arrangement of"  
750 PRINT "elements. These elements are shown below, in the order"  
760 PRINT "they appear in an actual program line"  
770 PRINT  
780 PRINT "STATEMENT (or LINE) NUMBER"  
790 PRINT  
800 PRINT "- indicates the processing sequence of the statements"  
810 PRINT  
820 PRINT "always in ascending order."  
830 PRINT "BASIC WORD"  
840 PRINT  
850 PRINT "specifies the computer operation to be performed"  
860 PRINT "PARAMETERS"  
870 PRINT  
880 PRINT "- variables, or expressions."  
890 PRINT  
900 PRINT "used to direct the operation performed"  
910 PRINT  
920 PRINT "the statement."  
930 PRINT  
940 INPUT "press ENTER to continue"  
950 GOSUB 9970
**** Listing of Program 'LESSON' ****

100 PRINT* STATEMENTS and PROGRAMS (cont.)
110 PRINT
120 PRINT "C150. PRINT*er. statement must have a line number and these numbers range"
130 PRINT "from 0 to 999 in most microcomputers that support"
140 PRINT "C151. PRINT"Microsoft BASIC" Microsoft is the company that owns"
150 PRINT "to the particular dialect of BASIC that"
160 PRINT "we're using on this machine. It is advisable to write program"
170 PRINT "lines in increments of 10 to allow you to insert additional"
180 PRINT "lines without having to renumber every statement line."
190 PRINT "C152. PRINT"the statements are executed in ascending numerical order."
200 PRINT "C153. PRINT"not in the order they were entered."
210 PRINT
220 PRINT "C154. PRINT*"look up the RENUM or NAME command in your manual for more."
230 PRINT "C155. PRINT*"information."
240 PRINT
250 PRINT
260 INPUT press ENTER to continue: T$%*%
270 GOSUB ???%
280 PRINT* STATEMENTS and PROGRAMS (cont.)
290 PRINT
300 PRINT "C156. PRINT"the last statement of the program should be the END statement."
310 PRINT "C157. PRINT"This indicates that the program is complete. IT IS NOT"
320 PRINT "C158. PRINT"NECESSARY, but it is a good practice to always put it in."
330 PRINT
340 PRINT "C159. PRINT"to get the program to execute you use the RUN command."
350 PRINT
360 PRINT "C160. PRINT"Now for a few questions to see how you are doing."
370 INPUT press ENTER for the questions: T$%
380 GOSUB ???%
390 PRINT* "A computer program is a series of - ------
400 PRINT
410 PRINT "C161. PRINT"A Verbs"
420 PRINT "C162. PRINT"Words"
430 PRINT "C163. PRINT"Statements"
440 PRINT "C164. PRINT"Syntaces"
450 PRINT
460 PRINT
470 INPUT "Be in the latter opposite the correct answer and press ENTER": T$%
480 PRINT
490 IF T$ = "C" GOTO 55726
500 PRINT* "CONGS - the correct answer is T. STATEMENTS."
510 GOTO 55725
520 PRINT "C165. PRINT*CORRECT"
530 PRINT
540 INPUT press ENTER to continue: T$%
550 GOSUB ???

530 PRINT "Which of the following is incorrect?"
530 PRINT
540 PRINT "A BASIC word is a word that a BASIC processor understands"  
550 PRINT "A statement can have no more than two line numbers"
560 PRINT "Data are the recipients of the action of BASIC verbs"
570 PRINT "Line numbers are written sequentially."
580 PRINT
590 PRINT "Press the letter that is beside the correct answer"  
600 PRINT "and then press ENTER."
610 PRINT
620 INPUT "What is your choice?"  
630 PRINT
640 IF T = "B" goto 670
650 PRINT "WRONG - Only one line number is allowed per statement"  
660 GOTO 650
670 PRINT "CORRECT - Way to go!"
680 PRINT
690 PRINT "Press the letter beside the correct answer and then press ENTER."
700 PRINT
710 INPUT "What is your choice?"
720 IF T = "A" GOTO 750
730 PRINT "WRONG - Data are too big. Remember you can only go to 5760."
740 GOTO 570
750 PRINT "CORRECT"
760 PRINT
770 INPUT "Press ENTER for the next question."
780 GOSUB 9970
790 PRINT "Is 35000 a valid statement number in Turbo BASIC?"
800 PRINT
810 PRINT "No"
820 PRINT "Yes"
830 PRINT
840 INPUT "Press the letter beside the correct answer and then press ENTER."
850 PRINT
860 IF T = "A" GOTO 890
870 PRINT "WRONG - 35000 is too big. Remember you can only go to 5680."
880 GOTO 570
890 PRINT "CORRECT"
900 PRINT
910 INPUT "Press ENTER."
920 GOSUB 10210
930 IF T = "A" GOTO 4800
940 RETURN
950 GOSUB 9970
960 PRINT "PRINT STATEMENT"
970 PRINT
980 PRINT "The BASIC word PRINT is a command that tells the computer"  
990 PRINT "to output the data that follows to the computer terminal"  
1000 PRINT
1010 PRINT "This data can be numbers, variables, or strings."  
1020 PRINT "Strings are combinations of words or numbers that are"  
1030 PRINT "to be printed without having any calculations done to them."
1040 PRINT
**** Listing of Program 'LESSON1' ****

5000 PRINT
5070 PRINT
5080 INPUT"press ENTER":"$
5090 GOSUB 9990
5100 PRINT"You can control the output caused by the print statement."
5110 PRINT"In two ways. If you just want what you ENTER printed"
5120 PRINT"without any calculations done to it, then you enclose the"
5130 PRINT"data after the PRINT command in quotation marks."
5140 PRINT
5150 PRINT"For Example:"$
5160 PRINT
5170 PRINT"PRINT CHR$(14)"So For It"CHR$(34):"
5180 PRINT"20 END"
5190 PRINT"p$"$
5200 PRINT
5210 PRINT"In BASIC, if you type this in as shown, you get this result:"$
5220 PRINT
5230 PRINT"So For It"
5240 PRINT
5250 INPUT"press ENTER":"$
5260 GOSUB 9990
5270 PRINT"Another example would be:"$
5280 PRINT
5290 PRINT"PRINT CHR$(34)"This is easy"CHR$(34):"
5300 PRINT"20 END"
5310 PRINT"RUN"
5320 PRINT
5330 PRINT"Which would result in:"$
5340 PRINT
5350 PRINT"This is easy"
5360 PRINT
5370 PRINT"Notice that nothing is changed by the computer, the words"
5380 PRINT"that were commanded to be output were printed exactly as shown."$
5390 PRINT
5400 INPUT"press ENTER for the second example":"$
5410 GOSUB 9990
5420 PRINT"The second way the print statement is used to control output."
5430 PRINT"is by NOT enclosing the data in quotation marks. The data is"
5440 PRINT"then read by the computer and the computer tries to evaluate"
5450 PRINT"what the data means in mathematical terms. If you have entered"
5460 PRINT"data that cannot be mathematically manipulated, then you"
5470 PRINT"get an ERROR message."

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

6520 FOR I = 1 TO 7
6530 PRINT
6540 NEXT I
6550 INPUT "press ENTER":I$
6560 GOSUB 9970
6570 PRINT
6580 PRINT "Here is an example of data in quotes, a data not in quotes"
6590 PRINT "and what the output would look like;"
6600 PRINT
6610 PRINT
6620 PRINT 10 PRINT CHR$(14)"I"+CHR$(14)
6630 PRINT "I"=I$ 20 PRINT I=I$
6640 PRINT "RUN"
6650 PRINT
6660 PRINT "Results:"
6670 PRINT "I-1"
6680 PRINT "I+1"
6690 PRINT
6700 INPUT C$:"$"
6710 PRINT
6720 INPUT "press ENTER":I$
6730 GOSUB 9970
6740 PRINT "PRINT (cont)"
6750 PRINT
6760 PRINT 10 PRINT CHR$(14)"I-1"+CHR$(14)
6770 PRINT "I"+CHR$(14)
6780 PRINT 20 PRINT I+I$
6790 PRINT "I-1"
6800 PRINT "I+1"
6810 PRINT
6820 PRINT "Note that the statement that had quotes was/repeated exactly"
6830 PRINT "as it was typed in, without the quotes, while the second"
6840 PRINT "statement was computed mathematically and a result was given."
6850 PRINT "The part of the first statement within quotes is called"
6860 PRINT "a STRING - 'remember'!"
6870 PRINT
6880 INPUT C$:"$"
6890 GOSUB 9970
6900 PRINT "PRINT (cont)"
6910 PRINT
6920 PRINT "The format of the output of PRINT can be controlled using"
6930 PRINT "commas. For Example:" 20 PRINT
6940 PRINT 10 PRINT CHR$(14)"A"+CHR$(14).6+I$+CHR$(14)"B"+CHR$(14)"C"
6950 PRINT 20 END
6960 PRINT"RUN"
*** Listing of Program 'LESSON1' ***

6980 PRINT
6990 PRINT "Results in:" 7000 PRINT
7010 PRINT "A",a+2,"B",b-1
7020 PRINT
7030 PRINT
7040 PRINT
7050 INPUT "press ENTER":T$ 7060 GOSUB 9970
7070 PRINT "A",a+2,"B",b-1 7080 PRINT
7090 PRINT "Notice how the commas have caused the terms of the"
7100 PRINT "statement to be spaced across the screen. The spaces"
7110 PRINT "are similar to TABS on a typewriter, however, the comma reacts"
7120 PRINT "differently on different terminals. See your BASIC manual."
7130 PRINT "or ask your system operator how they react on your machine."
7140 PRINT
7150 PRINT "commas usually cause 9 spaces between terms!"
7160 PRINT
7170 INPUT "press ENTER":T$ 7180 GOSUB 9970
7190 PRINT "PRINT (cont)"
7200 PRINT
7210 PRINT "The PRINT statement also allows you to output blank lines."
7220 PRINT "You print blank lines by typing in the line number and then"
7230 PRINT "a PRINT statement without an argument. For example:" 7240 PRINT
7250 PRINT "10 PRINT "CHR$(124)" Now is the time to skip"CHR$(124)
7260 PRINT "20 PRINT"
7270 PRINT "20 PRINT "CHR$(124)" a line."CHR$(124)
7280 PRINT "20 PRINT "a line"
7290 PRINT
7300 INPUT "press ENTER for results":T$ 7310 PRINT
7320 PRINT "Now is the time to skip"
7330 PRINT
7340 PRINT "a line"
7350 PRINT
7360 INPUT "press ENTER":T$ 7370 GOSUB 9970
7380 PRINT "12:10"
7390 PRINT
7400 PRINT "Which of the following statements would cause the above output?"
7410 PRINT
7420 PRINT "CHR$(124)"CHR$(124)
7430 PRINT

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

"140 PRINT"  20 PRINT 12+10"
"145 PRINT"
"150 PRINT"  15 PRINT "CHR$(24)*TEN+CHR$(24)"
"155 PRINT"
"160 PRINT"  25 PRINT 12+10"
"165 PRINT"
"170 INPUT"  press the letter that is beside the correct choice and ENTER":T$
"175 PRINT"
"180 IF $ = "A" GOTO 7630"
"185 PRINT"  "WRONG - the correct answer is A"
"190 PRINT"
"195 PRINT"  10 PRINT "CHR$(14)*12+10*CHR$(24)"
"200 PRINT"  "RUN"
"205 PRINT"
"210 PRINT"  "Which results in:"
"215 PRINT"
"220 PRINT"  12+10"
"225 PRINT"
"230 GOTO 7640"
"235 PRINT"  "Correct - that was an easy concept, congratulations!"
"240 PRINT"
"245 INPUT"  press ENTER":T$
"250 GOSUB 7979"
"255 PRINT"  "Write the statement that would cause a blank line to be printed"
"260 PRINT"  "Use 10 for the line number and leave one blank space between"
"265 PRINT"  "Terms.*"
"270 PRINT"
"275 INPUT"  what is your answer":T$
"280 PRINT"
"285 IF $ = "10 PRINT"  GOTO 7760"
"290 PRINT"  "WRONG - the correct answer is: 10 PRINT"
"295 GOTO 7770"
"300 PRINT"  "CORRECT"
"305 PRINT"
"310 INPUT"  press ENTER to continue":T$
"315 GOSUB 10210"
"320 IF $ = "B" GOTO 5950"
"325 RETURN"
"330 GOSUB 9979"
"335 PRINT"  "END & STOP"
"340 PRINT"
"345 PRINT"  "The END statement is the last statement in a program."
"350 PRINT"  "It notifies the computer when the program is done."
"355 PRINT"  "Because it is the last statement, it has the highest"
"360 PRINT"  "line number. The END statement is not necessary in Microsoft"
"365 PRINT"  "BASIC, but many programmers use it anyway. They believe a"
7900 PRINT"Program is more understandable and easier to "track" by"
7910 PRINT"another programmer if there is only ONE entry and ONE exit in"
7920 PRINT"a program."
7930 PRINT
7940 PRINT
7950 INPUT press ENTER for the rest of END & STOP";T#
7960 GOSUB 9970
7970 PRINT"The STOP statement interrupts execution of the program."
7980 PRINT"It is primarily used as a debugging aid. If you want"
7990 PRINT"to find the status of a variable at a certain point in a"
8000 PRINT"program, you insert a STOP statement. For example:"  
8010 PRINT
8020 PRINT 10 X = 2+3
8030 PRINT 20 Y = 1/5
8040 PRINT 30 STOP
8050 PRINT 40 X = Y+2
8060 PRINT
8070 PRINT"When this program is RUN it will STOP execution at line 30"
8080 PRINT"Then you may ask the computer to tell you the status of any"
8090 PRINT"of the variables X or Y. You can do this using the IMMEDIATE"  
8100 PRINT"mode (explained next section). Simply type in PRINT X,Y."
8110 PRINT
8120 PRINT press ENTER":T#$
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 = 125
8190 PRINT 20 Y = 242
8200 PRINT 30 STOP
8210 PRINT 40 PRINT X+Y
8220 PRINT"RUN"
8230 PRINT
8240 PRINT"Results in:"  
8250 PRINT
8260 PRINT 10 PRINT"BREAK IN 30"
8270 PRINT
8280 PRINT
8290 INPUT"hit ENTER for the rest":T#
8300 GOSUB 9970
8310 PRINT"BREAK in 30"
8320 PRINT
8330 PRINT
8340 PRINT"Now, by typing in CONT, the program will resume execution"  
8350 PRINT"For example:"
Listing of Program 'LESSON1'

3260 PRINT
3270 PRINT"CONT"*
3280 PRINT"22"
3290 PRINT
3400 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 TS = "B" GOTO 7820
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:"n3600 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:"n3710 PRINT
3720 PRINT"PRINT 9C+10+40"
3730 PRINT
3740 PRINT"Which would result in:"n3750 PRINT
3760 PRINT"143"
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."n3800 PRINT
3810 PRINT

59
***** Listing of Program 'LESSON1' *****

3920 PRINT
8830 INPUT"press ENTER";T$
3940 GOSUB 9970
3950 PRINT"IMMEDIATE (cont)"
3960 PRINT
3970 PRINT"The biggest disadvantage of the IMMEDIATE mode is that the data"
3980 PRINT"is not stored in memory, and cannot be repeated again."
3990 PRINT"It is lost after the initial display, whereas the programs we"
4000 PRINT"looked at before can be run over and over again by merely"
4010 PRINT"typing in the word RUN. Also, the IMMEDIATE mode is limited to"
4020 PRINT"one line of statements at a time."
4030 PRINT
4040 PRINT
4050 INPUT"press ENTER";T$
4060 GOSUB 9970
4070 PRINT"NEW Statement"
4080 PRINT
4090 PRINT"If you want to clean out the temporary memory in BASIC, all"
4100 PRINT"you have to do is type in the word NEW. BUT BE CAREFUL WITH"
4110 PRINT"THIS COMMAND. It will erase any program you have resident."
4120 PRINT
4130 PRINT
4140 PRINT"Let's say you've been practicing the commands you have learned"
4150 PRINT"so far, and you have put in a lot of line numbers and RUN them."
4160 PRINT"But now you want to start over. You can erase the mess with"
4170 PRINT"the command NEW."
4180 PRINT
4190 PRINT
4200 INPUT"press ENTER";T$
4210 GOSUB 9970
4220 PRINT"LIST"
4230 PRINT
4240 PRINT"Suppose you don't know what's there and you want to find out""
4250 PRINT"Just type in the command LIST. LIST will show you everything""
4260 PRINT"That's in temporary memory."
4270 PRINT
4280 PRINT
4290 INPUT"press ENTER";T$
4300 GOSUB 9970
4310 PRINT"DELETE"
4320 PRINT
4330 PRINT"Finally, what if you don't want to type in a new program."
4340 PRINT"you just want to delete a line? You can do that by typing in"
4350 PRINT"DELETE 10, or DELETE 10 or DELETE (line number)."
4360 PRINT
4370 PRINT"If you want to delete a range of line numbers, you type in""
4380 PRINT"DELETE (low range-high range). Say you want to delete lines""
4390 PRINT"15 to 35. You would type in DELETE 15-35, and the lines"
*** Listing of Program 'LESSON1' ***

"286 PRINT"would no longer be in memory"
289 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 9490
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 Program 'LESSON I'

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 9990
9880 PRINT "RIGHT AGAIN"
9890 PRINT
9900 INPUT "press ENTER" : T$
9910 GOSUB 10210
9920 IF T$ = "B" GOTO 8500
9930 RETURN
9940 REM ** This subroutine clears the screen on any terminal
9950 REM **
9960 FOR I = 1 TO 24
9970 PRINT
9980 NEXT I
9990 RETURN
10000 REM **
10010 REM ** This subroutine is the menu
10020 REM **
10030 PRINT " This is the first part of a two part lesson" : T$
10040 PRINT "It is divided into the following sections."
10050 PRINT
10060 PRINT 1) Introduction
10070 PRINT 2) Hardware
10080 PRINT 3) Software
10090 PRINT 4) General Information
10100 PRINT 5) Statements & Programs
10110 PRINT 6) Print Statement
10120 PRINT 7) End & Stop Statement
10130 PRINT 8) Immediate Mode, NEW
10140 PRINT
10150 RETURN
10160 RUN "MENU"
10170 RUN "LESSON I"
10180 REM **
10190 REM ** This subroutine lets student review sections again
10200 REM **
10210 GOSUB 9970
10220 PRINT"Which do you wish to do"
10230 PRINT
10240 PRINT"A Continue on"
10250 PRINT"B Review this section again"
10260 PRINT
10270 INPUT"press the letter opposite your choice and press ENTER";T$
10280 RETURN
1250 GOSUB 6960
1260 GOSUB 7000
1270 PRINT "A I'm taking this part in its entirety."
1280 PRINT "B I wish to review selected areas (or take the test)."
1290 PRINT "C I want to go to the first part."
1300 PRINT "D I want to return to the Menu."
1310 PRINT
1320 INPUT "Press either capital A, B, C, or D and then press ENTER": 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 3080
1400 GOSUB 4380
1410 GOSUB 5170
1420 GOSUB 6400
1430 GOSUB 6960
1440 GOSUB 7000
1450 PRINT
1460 PRINT "Please type in the number beside the area you wish:
1470 PRINT "to review (1 through 7) and then press ENTER - press 0 and"
1480 PRINT "press ENTER to return to the Menu."
1490 PRINT
1500 INPUT "What is your choice": N
1510 IF N = 0 GOTO 7220
1520 IF N = 7 GOTO 7210
1530 ON N GOSUB 1550,2570,3080,4380,5170,6400
1540 GOTO 1430
1550 GOSUB 6960
1560 PRINT "LIBRARY FUNCTIONS"
1570 PRINT
1580 PRINT "Many mathematical FUNCTIONS such as square root, trigonometric"
1590 PRINT "functions, and logarithms are difficult to derive using just"
1600 PRINT "addition, subtraction, multiplication, and division. To help"
1610 PRINT "us use these FUNCTIONS without deriving them from scratch"
1620 PRINT "each time we want to get a tangent or sine or square root, etc."
1630 PRINT "Microsoft BASIC has a library of commonly used FUNCTIONS"
1640 PRINT "already programmed into permanent memory. All you have to"
1650 PRINT "do is call them with a BASIC command whenever you want to use"
1660 PRINT "them."
1570 PRINT
1580 PRINT "You identify which function you want to use by using a keyword:" "
1590 PRINT "such as SQRT for square root."
1600 PRINT

**** Listing of Program 'LESSONIA' ****

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

```
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(25)*100'
1950 PRINT 'RUN'
1960 PRINT 'Which would give you:
1970 PRINT
1980 PRINT '500'
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'
```

*** Listing of Program 'LESSONIA' ***

2170 PRINT "program it was created in. We will discuss USER DEFINED"
2180 PRINT "FUNCTIONS in a later lesson."
2190 PRINT
2200 PRINT
2210 INPUT "press ENTER"; T$
2220 GOSUB 6960
2230 PRINT "QUIZ time! - USE ONLY CAPITAL LETTERS IN YOUR ANSWERS!"
2240 PRINT
2250 PRINT "Are the library functions stored permanently in memory?"
2260 PRINT
2270 PRINT "A Yes"; T$
2280 PRINT "B No"; T$
2290 PRINT
2300 INPUT "press the letter opposite the correct answer and press ENTER"; T$
2310 PRINT
2320 IF T$ = "A" THEN GOTO 2350
2330 IF T$ = "A" THEN PRINT "WRONG - the correct answer is A (Yes)!"
2340 GOTO 2360
2350 PRINT "CORRECT"
2360 PRINT
2370 INPUT "press ENTER to continue"; T$
2380 GOSUB 6960
2390 PRINT "If the function for converting a number to an integer is:
2400 PRINT "INT, show the statement for finding the integer value of"
2410 PRINT "27.56. Use line number 10, and leave only one space"
2420 PRINT "between elements. Do not include the RUN command."
2430 PRINT
2440 PRINT "DON'T FORGET TO ENCLOSE "27.56" IN 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 1550
2560 RETURN
2570 GOSUB 6960
1580 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"
LISTING OF PROGRAM 'LESSONIA' 07/10/83 - 01:59:45

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 4940 2750 PRINT 2760 PRINT "The way we treated letters in algebra, that is, assigning" 2770 PRINT "the values that were variable and were for calculation" 2780 PRINT "purposes, is the same way we treat them in the computer world." 2790 PRINT 2800 PRINT "For instance, if we give X the value of 8, then the computer" 2810 PRINT "will store the value 8 in a memory location that is labeled X." 2820 PRINT "The value will not change until we assign a new value to the" 2830 PRINT "label X, or quit BASIC." 2840 PRINT 2850 PRINT "There are two fundamental types of variables in BASIC, they" 2860 PRINT "are NUMERIC variables, and STRING variables. Our previous" 2870 PRINT "example of assigning a number made it a NUMERIC variable." 2880 PRINT 2890 INPUT "press ENTER": T$ 2900 GOSUB 4940 2910 PRINT "If we had assigned a CHARACTER (such as my name, DAN) to a" 2920 PRINT "variable, then we would have created a STRING variable." 2930 PRINT 2940 PRINT "A STRING variable holds data that will not be operated on" 2950 PRINT "mathematically. If I wouldn't want my name operated on, would" 2960 PRINT "you?" 2970 PRINT "The reason for having STRING variables is so we can do things" 2980 PRINT "like print labels, make word processors, and develop computer" 2990 PRINT "assisted instruction programs. If these tasks are done in" 3000 PRINT "BASIC, then they are done using STRINGS." 3010 PRINT 3020 INPUT "press ENTER": T$ 3030 GOSUB 7120 3040 IF T$ = "B" GOTO 2570 3050 RETURN 3060 GOSUB 3950
3090 "PRINT* Numeric Variables"
3100 PRINT
3110 PRINT*',In computers we assign values to variables to ease our job:"
3120 PRINT
3130 PRINT,'=1+2"
3140 PRINT
3150 PRINT*,In this case, the value of 3 would be assigned to X and the"
3160 PRINT*,'computer would store the value in its memory until we either"
3170 PRINT*,'changed it, or quit BASIC. In other words, we assigned the"
3180 PRINT*,'value of 3 to X, but only temporarily. Take the following"
3190 PRINT*,'example:"n
3200 PRINT
3210 PRINT,'10 X = 1+2"
3220 PRINT,'20 X = 4"
3230 PRINT
3240 PRINT*,What do you think the value of X is if we RUN the example?"
3250 PRINT
3260 INPUT*,'press ENTER for the answer':T$ *
3270 GOSUB 6960
3280 PRINT* Numeric Variables (cont)"
3290 PRINT
3300 PRINT*,Of course, you knew the answer was 4, didn't you?"
3310 PRINT
3320 PRINT*,Because long programs sometimes need many variables, Microsoft"
3330 PRINT*,'BASIC allows you to use all the letters of the alphabet PLUS"
3340 PRINT*,'it allows you to add a SECOND letter OR number to a variable"
3350 PRINT*,'to distinguish it from another. AI, X2, Y7, Z1, and FF are"
3360 PRINT*,legal variables. IA, ZZ, or Z1 are not legal. Can you see"
3370 PRINT*,'why? Right, they do not begin with a letter of the alphabet!"
3380 PRINT*,'(2 letters or 1 letter and 1 number are max length allowed)"
3390 PRINT
3400 PRINT*,you may also assign a value to a variable that is assigned to"
3410 PRINT*,'other variables. For instance:
3420 PRINT
3430 INPUT*,'press ENTER for an example of variable assignment':T$ *
3440 GOSUB 6960
3450 PRINT* 10 I = 4'
3460 PRINT* 20 Y = 7'
3470 PRINT* 30 Z = I-Y"
3480 PRINT
3490 PRINT*,'The variable Z is assigned the value of I+Y or 11:"
3500 PRINT
3510 PRINT
3520 PRINT*,'All variables are assigned the value of 0 when you first"
3530 PRINT*,'start up Microsoft BASIC. However, some languages assign"
3540 PRINT*,'indefinite values to all variables at first, and wait for you'
3550 PRINT "to change them. That is why you say"
3560 PRINT "a variable to 0 when there appears to be no other reason for it."
3570 PRINT
3580 PRINT
3590 PRINT
3600 INPUT "press ENTER"; T$
3610 GOSUB 6960
3620 PRINT * Numeric Variables (cont) *
3630 PRINT
3640 PRINT "When you use variables on the right side of an equation"
3650 PRINT "you must have assigned values to the variables previously. It's"
3660 PRINT "a KEY CONCEPT that the equal sign does not mean mathematical"
3670 PRINT "equality. The equal sign is an ASSIGNMENT statement."
3680 PRINT "It ASSIGNs the value on the right side of the equation"
3690 PRINT "to the variable on the left."
3700 PRINT
3710 PRINT 10 X = X$2*
3720 PRINT
3730 PRINT "In the above statement, X will be assigned the value of "
3740 PRINT X$2, or 2"
3750 PRINT
3760 INPUT "press ENTER"; T$
3770 GOSUB 6960
3780 PRINT * Numeric Variables (cont) *
3790 PRINT
3800 PRINT 10 X = 4*
3810 PRINT
3820 PRINT "In the above example, we assigned the value of 4 to X."
3830 PRINT "In some dialects of BASIC, we must use the word LET to assign"
3840 PRINT "a value to a variable."
3850 PRINT
3860 PRINT 10 LET X = 4*
3870 PRINT
3880 PRINT "Such as above. It is not necessary to use the word LET in"
3890 PRINT "Microsoft BASIC. We only mention it because you may wish to"
3900 PRINT "copy a program written in another dialect onto Microsoft."
3910 PRINT "you do, you may either leave the LET word in or arco it, the"
3920 PRINT "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 A = 2"
3990 PRINT "B 20 X = 2$"
4000 PRINT "C 15 XYZ = 56"
**** Listing of Program 'LESSONIA' ****

400 PRINT"ED 10 23 = A+B"
403 PRINT
407 INPUT"press the letter opposite the correct answer and press ENTER":$1
409 PRINT
412 IF $1 = "A" SOTO 4120
414 PRINT"Wrong - the correct answer is A ( LET A = 2"
416 PRINT
418 PRINT(This is a KEY concept. You may wish to review variables"
420 PRINT"again, to be sure you understand them."
422 PRINT
425 SOTO 4140
427 PRINT "CORRECT - Good job!!"
429 PRINT
432 INPUT: press ENTER: $1
434 SUB $930
436 PRINT"which of the following is a legal statement if variables A and"
438 PRINT"B have previously been assigned a value?"
440 PRINT
443 PRINTA 10 A+2 = C"
445 PRINTA 10 D = A-B"     
447 PRINTC 10 "22 = A+B"
449 PRINTD 10 222 = A+3"
451 PRINT
454 PRINT"press the letter opposite the correct answer":$1
457 PRINT
459 IF $1 = "S" SOTO 4520
461 PRINT"Wrong - the correct answer is C = A+B"
463 PRINT"You may need to review this section if you did not get this"
465 PRINT"Question right."
467 PRINT
470 SOTO 4140
472 PRINT"CORRECT - that was a KEY concept. You're doing good!!"
474 PRINT
476 INPUT: press ENTER: $1
478 SUB $130
480 IF $1 = "B" SOTO 5060
482 RETURN
485 SOTO 5530
487 PRINT"String Variables"
490 PRINT
492 PRINT"Whenever you assign CHARACTERS or NUMBERS that will not be"
494 PRINT"mathematically manipulated - such as a street address to a"
496 PRINT"variable, you have created a STRING variable."
498 PRINT
500 PRINT"There is a special way of making a STRING in BASIC. You "DEF"
502 PRINT"attach a dollar sign ($) to the end of a variable label."
Listing of Program "LESSON1A"

447: PRINT "When you do, the computer will know that this is a STRING and"
448: PRINT "will not try to manipulate it. In addition, everything that"
449: PRINT "you want to be included in the string must be enclosed in"
450: PRINT "quotes. For example:""""""
451: PRINT
452: PRINT **$1 = "CHAR$(34)"The author is "CHAR$(34)""
453: PRINT
454: INPUT "press ENTER":$1
455: GOSUB 2950
456: PRINT "String Variables (cont):"
457: PRINT
458: PRINT **$1 = "CHAR$(34)"The author is "CHAR$(34)"
459: PRINT
460: PRINT "Here, the variable label L is identified as a STRING variable"
461: PRINT "by the addition of a dollar sign. Further, the CHARACTER data"
462: PRINT ""CHR$(34)" is identified as a STRING label.""
463: PRINT
464: PRINT **$1 = "CHAR$(34)"114 west Cottage Street"CHR$(34)"
465: PRINT
466: PRINT "In this example, we have assigned both letters and numbers to"
467: PRINT "the STRING."""
468: PRINT
469: PRINT "Examples of illegal STRING labels would be 6, 114, 1214, JIM:""
470: PRINT
471: INPUT "press ENTER":$1
472: GOSUB 2950
473: PRINT **$1 = "CHR$(34)"The author is "CHR$(34)"
474: PRINT
475: PRINT "which of the following are correct STRINGS?:"
476: PRINT
477: PRINT **$1 = "AT $1 = 12"
478: PRINT **$1 = "CHR$(34)"CHR$(34)"
479: PRINT **$1 = "North State Street"$1 = "North State Street"
480: PRINT
481: INPUT "press the letter opposite the correct answer and press ENTER":$1
482: PRINT
483: IF $1 = "G" GOTO 4950
484: PRINT "Wrong - the correct answer is 6 $1 = "CHR$(34)"CHR$(34)"
485: GOTO 4950
486: PRINT "Correct - are you sure you are only a student?"
487: PRINT
488: INPUT "press ENTER":$1
489: GOTO 4950
490: PRINT **$1 = "CHR$(34)" This is really awesome, I swear!"CHR$(34)"
491: PRINT
492: PRINT **$1 = "CHR$(34)" This is really awesome, I swear!"CHR$(34)"
493: PRINT
494: PRINT **$1 = "CHR$(34)" This is really awesome, I swear!"CHR$(34)"
495: PRINT
496: PRINT **$1 = "CHR$(34)" This is really awesome, I swear!"CHR$(34)"
497: PRINT
498: PRINT **$1 = "CHR$(34)" This is really awesome, I swear!"CHR$(34)"
499: PRINT
500: PRINT **$1 = "CHR$(34)" This is really awesome, I swear!"CHR$(34)"
***** Listing of Program 'LESSONIA' *****

100 PRINT "RUN"
110 PRINT
120 PRINT " This is really, umm, I mean, really."
130 PRINT
140 PRINT "Do you see how the computer treated the data?"
150 PRINT "What would the output of this program?"
160 PRINT
170 PRINT 10: DATA = "CHAR.34: " "THIS IS A TEST QUESTION" CHAR.34:
180 PRINT RUN
190 PRINT
200 PRINT Type in the correct answer exactly as it would be printed"
210 INPUT 'a'
220 PRINT
230 IF T = 'THIS IS A TEST QUESTION' THEN 510
240 PRINT "WRONG - the correct answer is!"
250 PRINT
260 PRINT 'THIS IS A TEST QUESTION'
270 DATA 300, 111
280 PRINT 'CORRECT'
290 PRINT
300 PRINT 'Press ENTER to:
310 INPUT 'press ENTER to:
320 GOTO 280: "12."
330 IF T = '2' GOTO 430
340 RETURN
350 GOTO 280
360 PRINT "Using Arithmetic"
370 PRINT
380 PRINT "BASIC will let you use arithmetic to figure out almost anything".
390 PRINT "Mathematical tasks you would want. BASIC uses four operators to"
400 PRINT "perform addition, subtraction, multiplication, division and"
410 PRINT "exponentiation (raising something to a power). Here they are:
420 PRINT
430 PRINT "Symbol Meaning Example"
440 PRINT 150 PRINT " + addition A + B"
450 PRINT " - subtraction C - D"
460 PRINT " * multiplication E * F"
470 PRINT " / division G / H"
480 PRINT " ^ exponentiation M ^ 2 'A squared:'"
490 PRINT
500 PRINT "Parentheses - are also used, just as in algebra:
510 PRINT
520 PRINT 150 PRINT "Press ENTER to:
530 INPUT 'press ENTER to:
540 GOTO 420: "data"
550 PRINT "Using Arithmetic cont."
560 PRINT
570 PRINT
Listing of Program 'LESSONIA' 

PRINT"Note that a "CHAR" or "CHRS" always must be used for multiplication."
PRINT"- you tried to use an X or I, the computer"
PRINT"would think you were trying to put in another variable label."
PRINT"and would give you an error message. Also, you cannot use" " CHRS (F lieutenant) to mean 555, if you do, you will get an error." PRINT"if you put two variables together, like A and B, to make AB."
PRINT"You are not multiplying them, rather, you just created a NEW" PRINT"variable (AB)"
PRINT
PRINT"So always remember to use the asterisk for multiplication."
PRINT
INPUT"Press ENTER to continue" USING Arithmetic (cont)"
PRINT
PRINT "The symbols we just looked at are called ARITHMETIC OPERATIONS" PRINT "and they may be combined in any order in a BASIC statement."
PRINT "However, just like mathematics, the computer will treat some" PRINT "symbols with a higher priority than others. For example:" PRINT "SOLVE 1 = (3-2)*((2+7)*2)" PRINT
PRINT "This statement, the computer will scan the line and do" PRINT "all terms within parenthesis first. Then it" PRINT "will scan for exponentiation, before those operations, then" PRINT "it will scan for multiplication OR division and perform those" PRINT "operations as it comes to them, and finally... it will scan" PRINT "for addition OR subtraction and perform those operations." PRINT
INPUT"Press ENTER to continue" USING Arithmetic (cont)"
INPUT"Press ENTER to continue" USING Arithmetic (cont)"
PRINT
INPUT"Press ENTER to continue" USING Arithmetic (cont)"
PRINT
PRINT"The computer always scans from left to right. It will scan" PRINT"once for each category of symbols. The categories" PRINT"are restated below:" PRINT
PRINT"Category: Priority:" PRINT
PRINT"** HIGHEST**" PRINT
PRINT"* or NEXT HIGHEST**" PRINT
PRINT"+ or LOWEST" PRINT
PRINT
INPUT"Press ENTER for more" PRINT
INPUT"Press ENTER for more" PRINT
INPUT"Press ENTER for more" PRINT
INPUT"Press ENTER for more" PRINT
Using Arithmetic (cont.)

On the first scan, the computer would do the terms within the parenthesis. It would first do exponentiation (\(2^2=4\)). Then it would do the multiplication, and finally the addition.

Using Arithmetic

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

For example:

```plaintext
50 PRINT "B = 33*(2+1)"
60 PRINT
70 PRINT "is much different than:"
80 PRINT
90 PRINT "G=5*2;"
100 PRINT
110 PRINT "Do you see why? The first value assigned to B is 3997, the second value assigned is 1099. Study the example carefully."
```

INPUT "press ENTER to assign the value of 10" to the variable M

```
130 PRINT
140 PRINT "M = 10*2+2-3"
150 PRINT "M = 5*2+3*4+1-10" 
160 PRINT "M = 5*2+3*4+1-10"
170 PRINT "M = 10*2-3"
180 PRINT
190 PRINT "press the letter opposite the correct answer and press ENTER to continue"
```

If M = 33 GOTO 140

DIF M = 33 PRINT "Wrong - the right answer is 10#. M = 5*2+3*4+1-10"
**** Listing of Program "LESSON4" ****

110 PRINT "M = 5*E,(10) - 10"
120 PRINT "M = 10 + 10 - 10."
210 GOTO 670
220 PRINT "CORRECT!"
630 PRINT
639 INPUT "press ENTER to continue": T$
670 GOSUB 7120
679 IF T$ = "B" GOTO 6170
679 RETURN
640 GOSUB 960
6410 PRINT "Comparing Variables"
6420 PRINT
6430 PRINT "BASIC uses symbols to compare values to determine relationships"
6440 PRINT "such as whether one variable is less than, more than, or equal"
6450 PRINT "to another variable. We have already used one of these symbols"
6460 PRINT "= it is called the equal sign (=). When you start programming,"
6470 PRINT "you will often want to check to see if one variable is dif-
6480 PRINT "ferent than another. There are six symbols you can use to"
6490 PRINT "do this."
6500 PRINT
6510 INPUT "press ENTER for examples": T$
6520 GOSUB 960
6530 PRINT "Comparing Variables (cont.)"
6540 PRINT
6550 PRINT "Symbol table:"
6560 PRINT
6570 PRINT a580 PRINT * Symbol Meaning Example"
6590 PRINT "= equal A=2"
6610 PRINT "< less than A<B"
6620 PRINT "> greater than A>B"
6630 PRINT "<> not equal to A<>B"
6640 PRINT "<= less than or equal A<=B"
6650 PRINT ">= greater than or equal A>=B"
6663 PRINT
6670 PRINT "We will discuss these in more detail in a later lesson."
6680 PRINT
6690 INPUT "press ENTER": T$
6700 GOSUB 6960
6710 PRINT "That concludes this lesson. When you hit ENTER you will"
6720 PRINT "be returned to the start of this part. You may either take"
6730 PRINT "the test or review selected areas."
6740 PRINT
6750 PRINT "By now, you should understand the following program:"
### Listing of Program 'LESSONIA'

```
3790 PRINT"10 PRINT"CHR$(44)"The product of 10 times 3 is "CHR$(44)"
3795 PRINT"20 PRINT 10 * 3"
3800 PRINT"30 PRINT"
3805 PRINT"40 PRINT" END
3810 PRINT"RUN"

Listing of 'LESSONIA'

1) 71tO1V
01:59:45

a770 PRINT'10 PRINT"CHRS(43)"The product of 10 times 3 is "CHRS(43)"
630 PRINT'20 PRINT 10 * 3"
1440 PRINT'30 PRINT"
6790 PRINT'40 PRINT" ORINT" END

Listing of Program 'LESSONIA'

A850 PRINT'similar to this. Line 10 prints the string. Line 20 prints" 6800 PRINT'the mathematical calculation. Lines 30 and 40" 6870 PRINT"print two blank lines. You make the program work by using" 6880 PRINT"the RUN word after you have entered the statements."

Listing of Program 'LESSONIA'

7290 PRINT
7310 INPUT'press ENTER":IT$
7320 RUN
7330 REM **
7340 REM ** This subroutine clears the screen on any terminal
7350 REM **
7360 FOR I = 1 TO 24
7370 PRINT
7380 NEXT I
7390 RETURN
7400 PRINT"LESSON 18"

Listing of Program 'LESSONIA'

7420 PRINT
7430 PRINT"This is the second part of a two part lesson"
7440 PRINT"It is divided into the following sections."
7450 PRINT
7460 PRINT"Library Functions 1) String Variables"
7470 PRINT"Variables (general) 2) Using Arithmetic"
7480 PRINT"Numeric Variables 3) Comparing Variables"
7500 PRINT" & Lesson Summary"
7510 PRINT TEST
7520 RUN
7530 RETURN
7540 GOSUB 6790
7550 PRINT"Which do you want to do?"
7560 PRINT A" Continue on"
7570 PRINT B" Review this lesson again"
7580 PRINT
7590 INPUT'press the letter opposite your choice and press ENTER":IT$
7610 CF"S": A AND IT$: B GOTO 7620
7620 RETURN
7630 RUN "TEST1"
7640 RUN "MENU"
```
**** Listing of Program 'LESSON1A' ****

7250 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 ** NS(I) = NAMES ARRAY, USED TO READ IN SEQUENTIAL NAMES, AND TO WRITE OUT
1070 REM **
1080 REM ** UPDATE NAMES.
1090 REM ** S(I) = SCORES ARRAY - USED TO READ AND WRITE SCORES
1100 REM **
1110 REM ** Q(I) = ARRAY TO KEEP TRACK OF NUMBER OF CORRECT ANSWERS. IF AN ARRAY ELEMENT EQUALS 1, THE ANSWER WAS CORRECT
1120 REM **
1130 REM **
1140 CLEAR 3000
1150 DIM NS(1000)
1160 DIM S(10)
1170 DIM Q(1000)
1180 PRINT "FINAL TEST (lesson 1)"
1190 PRINT
1200 PRINT "This test consists of 10 questions, you must get 70 percent of them correct to pass. (that's 7 right out of the 10 questions). Use only capital letters in your answers, don't include extra spaces or letters. If you answer a question wrong, you get the correct answer, plus a reference for review. In addition, you will get a synopsis of areas for review at the end of the test."
1210 PRINT
1220 IF NS(I) = "A" THEN GOTO 1260
1230 PRINT "Wrong - the correct answer is A (Yes. programs are software)"
1240 PRINT "review part 1, hardware and software."
1250 PRINT
1260 PRINT "Yes"
1270 PRINT "No"
1280 PRINT "press the letter opposite the correct answer and press ENTER":?T$:?
1290 PRINT
1300 PRINT "If you successfully pass the test, you will be given your homework assignment. GOOD LUCK!"
1310 PRINT
1320 INPUT "press ENTER to continue":?T$:?
1330 PRINT "is a computer program called Software"
1340 PRINT
1350 GOSUB 4130
1360 IF T$ = "A" THEN GOTO 1370
1370 PRINT "Wrong - the correct answer is A (Yes. programs are software)"
1380 PRINT "review part 1, hardware and software."
**** Listing of Program 'TEST' ****

1460 GOTO 1490
1470 PRINT"CORRECT"
1480 O(1) = 1
1489 PRINT
1500 INPUT"press ENTER":T$
1510 GOSUB 4120
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":TS
1600 PRINT
1610 IF TS = "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 O(2) = 1
1690 PRINT
1700 INPUT"press ENTER to continue":T$
1710 GOSUB 4120
1720 PRINT"Which statement would print the word TEST"
1730 PRINT
1740 PRINT"A	PRINT TEST"
1750 PRINT"B	PRINT CHRS(34)""TEST""CHRS(34)"
1760 PRINT"C	OUTPUT CHRS(34)""TEST""CHRS(34)"
1770 PRINT"D	PRINT 'TEST'"
1780 PRINT
1790 INPUT"press the letter opposite the correct answer and press ENTER":TS
1800 PRINT
1810 IF TS = "A" THEN GOTO 1850
1820 PRINT"WRONG - the correct answer is B (PRINT CHRS(34)""TEST""CHRS(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 D uses the wrong characters for quotes."
1860 PRINT"review part 1, PRINT, and part 2, String Variables"
1870 GOTO 1850
1880 PRINT"CORRECT"
1890 Q(3) = 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 I = SQR(4) *"
1910 PRINT "20 PRINT 5*I"
1920 PRINT "RUN"
1930 PRINT
1940 :INPUT "Type in your answer and press ENTER":T$;
1950 PRINT
1960 IF T$ = "25" THEN GOTO 2000
1970 PRINT "Wrong - the correct answer is 25"
1972 PRINT line 10 puts the square root of 4 into the"
1974 PRINT "variable I, line 20 causes 5 to be taken to"
1976 PRINT "the power of 2. 5 squared is 25."
1980 PRINT "review part 1, PRINT, and part 2. FUNCTIONS"
1990 GOTO 2020
2000 PRINT "CORRECT"
2010 Q(4) = 1
2020 PRINT
2030 INPUT "press ENTER":T$
2040 GOSUB 4130
2050 PRINT "Give the necessary statement to print a blank line. Use"
2060 PRINT "line number 19 and leave one blank space between terms."
2070 PRINT
2080 INPUT "What's your answer":T$
2090 PRINT
2100 IF T$ = "10 PRINT" THEN GOTO 2140
2110 PRINT "Wrong - the correct answer is 10 PRINT"
2120 PRINT "review part 1, PRINT"
2130 GOTO 2160
2140 PRINT "CORRECT"
2150 Q(5) = 1
2160 PRINT
2170 INPUT "press ENTER":T$
2190 GOSUB 4130
2190 PRINT "The two types of FUNCTIONS are LIBRARY and COMPUTER, TRUE or"
2200 PRINT "FALSE"
2210 PRINT
2220 PRINT "A TRUE"
2230 PRINT "B FALSE"
2240 PRINT
2250 INPUT "press the letter opposite the correct answer and press ENTER":T$
2260 PRINT
2270 IF T$ = "B" THEN GOTO 2330
2280 PRINT "Wrong - the correct answer is B (False) - the two types"
2290 PRINT "of functions are LIBRARY and USER. Review part"
2300 PRINT* "2. FUNCTIONS."
2310 PRINT
2320 GOTO 2350
2330 PRINT"CORRECT"
2340 Q(1) = 1
2350 PRINT
2360 INPUT"press ENTER";T$
2370 GOSUB 4130
2380 PRINT"Which of the following statements is invalid?"
2390 PRINT
2400 PRINT"A 25 = X"
2410 PRINT"B PRINT SQR(25)"
2420 PRINT"C PRINT 25**10"
2430 PRINT"D L = M+N"
2440 PRINT
2450 INPUT"Type in the letter opposite the INCORRECT statement";T$
2460 PRINT
2470 IF T$ = "A" THEN GOTO 2510
2480 PRINT"WRONG - the INCORRECT statement is A (25 = X)"
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"memory. *TRUE or FALSE*"
2590 PRINT
2600 INPUT"Type in TRUE or type in FALSE for this statement";T$
2610 PRINT
2620 IF T$ = "FALSE" THEN GOTO 2670
2630 PRINT"WRONG - the answer is FALSE"
2640 PRINT"strings are not used for manipulating math"
2650 PRINT"equations. Review part 2, Strings."
2660 GOTO 2690
2670 PRINT"CORRECT"
2680 Q(8) = 1
2690 PRINT
2700 INPUT"press ENTER";T$
2710 GOSUB 4130
2720 PRINT"Which of the following is an example of a peripheral device?"
2730 PRINT
2740 PRINT"A CPU"
2750 PRINT"B MEMORY"
2760 PRINT"C KEYBOARD"
2770 PRINT
2780 INPUT"press the letter opposite the correct answer and press ENTER";T$
2790 PRINT
2800 IF T$ = "C" THEN GOTO 2840
2810 PRINT"WRONG - the correct answer is C (keyboard)"
2812 PRINT"Answers A & B are not peripherals, they are part of the main computer structure."
2820 PRINT"review part 1, General Information"
2830 GOTO 2860
2840 PRINT"CORRECT"
2850 G(9) = 1
2860 PRINT
2870 INPUT"press ENTER";T$
2880 GOSUB 4130
2890 PRINT"If you had the following program in memory;"
2900 PRINT
2910 PRINT"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"press the letter opposite the correct answer and press ENTER";T$
3040 PRINT
3050 IF T$ = "D" THEN GOTO 3090
3060 PRINT"WRONG - the correct answer is D (DELETE 20-30)"
3062 PRINT"Answer A would delete the whole program, and"
3064 PRINT"answer B prints the sum of 10 and 40. Answer C would cause a syntax error."
3070 PRINT"review part 1, IMMEDIATE, NEW, DELETE"
3080 GOTO 3110
3090 PRINT"CORRECT"
3100 G(10) = 1
3110 PRINT
3120 INPUT"press ENTER";T$
3130 GOSUB 4130
3140 FOR I = 1 TO 10
3150   Y = Y + G(I)
3160 NEXT I
3170 PRINT
**** Listing of Program 'TEST1' ****

160 NEXT X
170 PRINT "You have finished the test, out of 10 possible correct answers"
180 PRINT "You scored "Y,""
190 PRINT
200 IF Y > 6 THEN PRINT "YOU HAVE PASSED"
210 GOSUB 3950
220 IF Y > 6 THEN GOTO 3300
230 PRINT "YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
240 PRINT
250 PRINT "YOU SHOULD RETAKE LESSON 1 BEFORE GOING FARTHER!"
260 PRINT
270 PRINT "You will be returned to the Menu."
280 PRINT
290 GOTO 4170
300 PRINT
310 PRINT "Do you want your score recorded on a permanent file?"
320 PRINT
330 PRINT "A YES"
340 PRINT "B NO"
350 PRINT
360 INPUT "Which?"
370 IF T1 = "B" THEN GOTO 3600
380 GOSUB 4130
390 PRINT "To record your score, we must open a file and put your name"
400 PRINT "in it. Therefore, surprisingly, we need your name. If your"
410 PRINT "name is not unique among the students likely to take this test,"
420 PRINT "please contact your test monitor for an identifying word that"
430 PRINT "will make you unique. Then enter that word below."
440 PRINT
450 PRINT "If you have already entered a score previously, be sure to"
460 PRINT "enter the same name you used before. Use all capitals."
470 PRINT
480 INPUT "ENTER your word or name now:"
490 OPEN #: 1 "SCORE1"
500 X = 0
510 IF EOF(): THEN GOTO 5570
520 X = X + 1
530 INPUT X.N$: X
540 INPUT #: X
550 IF N.X$ = "1" THEN GOTO 3B10
560 GOTO 5510
570 CLOSE
580 X = X
590 N.X$ = 1
600 S(X) = Y
610 OPEN #: 1 "SCORE1"
**** Listing of Program 'TEST1' ****

3620 FOR W = 1 TO I
3630 PRINT#1,W(W)
3640 PRINT#1,5(W)
3650 NEXT W
3660 GOSUB 4170
3670 PRINT"You are now qualified to go to LESSON 2."
3680 PRINT
3690 PRINT"Your homework assignment is:"
3700 PRINT
3710 PRINT"Write a short program that will state the following when RUN:"  
3720 PRINT
3730 PRINT"THE SUM OF 3, 2, AND Z IS"
3740 PRINT"27"
3750 PRINT
3760 GOTO 4170
3780 X(W) = Y
3790 IF EOF THEN CLOSE:GOTO 3860
3800 D = 1+1
3810 INPUT#1, X(W), S(X)
3820 OPEN#1,"SCORE1" FOR # = I TO X
3830 PRINT#1,4(W)
3840 NEXT W
3850 PRINT"You now take LESSON 2. You will be returned to the MENU"  
3860 PRINT"from where you may go to LESSON 2 or quit."
3870 GOTO 4170
3880 IF Y=10 THEN RETURN
3890 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
3900 PRINT
3910 IF 3(1) = 0 THEN PRINT* part 1, Hardware and Software"  
3920 IF 2(2) = 0 OR 44 = 0 OR 56 = 0 THEN PRINT* part 2, Functions"
3930 IF 2(2) = 0 THEN PRINT* part 1, Print, and part 2, String Variables"
3940 IF 2(2) = 0 THEN PRINT* part 1, Print*"
3950 IF 2(2) = 0 THEN PRINT* part 2, Variables*"
3960 IF 2(2) = 0 THEN PRINT* part 2, String Variables*"
3970 IF 2(2) = 0 THEN PRINT* part 1, General Information*"
3980 IF 2(2) = 0 THEN PRINT* part 1, IMMEDIATE*"

84
***** Listing of Program 'IESTI' *****

4060 PRINT
4070 INPUT"press ENTER";I$
4080 GDSUB 4110
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";I$
4190 RUN"MENU"
***** Listing of Program 'LESON2' *****

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 3120
1080 PRINT "LESSON:  BASIC 2   VERSION: 1 AUGUST 83"
1090 PRINT
1100 PRINT "TIME REQUIRED TO COMPLETE LESSON: About One Hour"
1110 PRINT
1120 PRINT
1130 PRINT "AUTHOR: Capt Danny J. Creagan"
1140 PRINT "          Air Force Institute of Technology"
1150 PRINT
1160 PRINT "OBJECTIVE: To teach the student about permanent storage."
1170 PRINT "          how to handle data in BASIC, and how to branch to"
1180 PRINT "          different parts of a program."
1190 PRINT
1200 PRINT
1210 PRINT
1220 PRINT
1230 PRINT "press the ENTER key to continue": T$ 1%
1240 GOSUB 8120
1250 GOSUB 6150
1260 PRINT "1. I'm taking this part in its entirety."
1270 PRINT "2. I wish to review selected areas."
1280 PRINT "3. I want to go to the second part."
1290 PRINT "4. I want to return to the menu."
1300 PRINT
1310 PRINT "Press any capital A, B, C, or D and then press ENTER": T$ 1%
1320 IF T$ = "D" GOTO 1260
1330 IF T$ = "C" GOTO 9370
1340 IF T$ = "B" GOTO 1430
1350 IF T$ = "A" GOTO 1260
1360 GOSUB 1540
1370 GOSUB 1950
1380 GOSUB 3870
1390 GOSUB 5000
1400 GOSUB 6570
1410 GOSUB 8120
1420 GOTO 9370
1430 GOSUB 8120
1440 GOSUB 8150
1450 PRINT
**** Listing of Program 'LESSON2' **** 07/10/83 - 00:01:19

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 1540,1950,3870,5060,5670
1530 GOTO 1420
1540 GOSUB 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 1600
1570 GOSUB 8120
1580 PRINT "Here is one possible way to complete your homework:"
1590 PRINT
1600 PRINT 10 PRINT CHR$(34) "THE SUM OF 3, 2, AND 22 IS" CHR$(34)
1610 PRINT 20 PRINT 3 + 2 + 22
1620 PRINT 30 PRINT 1610 PRINT 40 PRINT
1622 PRINT 40 PRINT "UN"
1623 PRINT "THE SUM OF 3, 2, AND 22 IS"
1625 PRINT "27"
1626 PRINT
1630 PRINT
1640 INPUT "You can try this one if you had trouble with yours. Press ENTER": T$  
1650 T$ = ""
1660 GOSUB 3120
1670 PRINT 50 PRINT "Introduction"
1680 PRINT
1690 PRINT "In this lesson we will cover some of the most exciting and"
1700 PRINT "useful commands in the BASIC language. When we left off,"  
1710 PRINT "in lesson 1, we had discussed some of the fundamental commands"  
1720 PRINT "that you must use just to get BASIC started. Now, we will"  
1730 PRINT "discover how to SAVE our programs for future use, how to"  
1740 PRINT "control data input in our program, and how to leave little"  
1750 PRINT "messages in our program so that other programmers can under-
1760 PRINT "stand what we are trying to do. Most importantly, we will"  
1770 PRINT "discover how to branch to different parts of a program"  
1780 PRINT "depending on our data manipulation requirements. That way, one"  
1790 PRINT "program can be extremely flexible and do many different kinds"  
1800 PRINT "of work for us."
1810 PRINT
1820 INPUT "Press ENTER to continue": T$
1830 GOSUB 3120
1840 PRINT 50 PRINT "Introduction"
1850 PRINT
1860 PRINT "After taking this lesson, we recommend you practice some of the"
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 "Filenames"
1970 PRINT
1980 PRINT "Remember, in lesson 1, when we discovered how to make programs"
1990 PRINT "that could be RUN over and over? We said then that the pro-
2000 PRINT "gram was stored in TEMPORARY memory. If you tried a few of"
2010 PRINT "the examples that were given, you will have noticed that the"
2020 PRINT "program was destroyed whenever you left BASIC. This section"
2030 PRINT "and the next section will show you how to SAVE a program, and"
2040 PRINT "then call it back from PERMANENT storage. That way, when"
2050 PRINT "you've spent hours making the best data manager ever written."
2060 PRINT "you won't have to re-write it when you turn on the machine"
2070 PRINT "again!"
2080 PRINT
2090 PRINT "PERMANENT storage is the way we store data for an indefinite"
2100 PRINT "period. We usually use DISKS or TAPE for PERMANENT storage."
2110 INPUT "press ENTER":T$
2120 GOSUB 8120
2130 PRINT "Filenames (cont)"
2140 PRINT
2150 PRINT "For the purposes of this lesson, we will assume you only use"
2160 PRINT "DISKS for permanent storage."
2170 PRINT
2180 PRINT "A DISK is a platter of iron-oxide coated material that stores"
2190 PRINT "data almost the same way that an audio tape stores music."
2200 PRINT "A DISK comes in many sizes and with many different storage"
2210 PRINT "capabilities. Fortunately, the way we store data on disk when"
2220 PRINT "we are using Microsoft BASIC is standardized for almost all"
2230 PRINT "installations. (there is a slight difference if you are using"
2240 PRINT "a TRS-80, we will explain it as we go along)"
2250 PRINT
2260 INPUT "press ENTER":T$
2270 GOSUB 8120
2280 PRINT "Filenames (cont)"
2290 PRINT
2300 PRINT "What happens when you store data? Well, the computer takes"
2310 PRINT "care of most of the details, it waits until you tell it to"
2320 PRINT "store a program, then it searches the available storage areas"
**Listing of Program 'LESSON2'**

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 "Filenames (cont)"
2420 PRINT
2430 PRINT "This label is called a FILENAME. FILENAMES are very strictly"
2440 PRINT "controlled by the computer. They must follow the following"
2450 PRINT "format EXACTLY."
2460 PRINT
2470 PRINT 
2480 PRINT "TRS-80 CROMEMCO (or CPM)"
2490 PRINT
2500 PRINT 
2510 PRINT "Notice that the only difference between a TRS-80 and CROMEMCO"
2520 PRINT "is that the TRS-80 has a slash, ‘/’, between the filename"
2530 PRINT "and the extension, while the CROMEMCO has a period or dot, ‘.’"
2540 PRINT
2550 INPUT "press ENTER":T$
2560 GOSUB 8120
2570 PRINT "Filenames (cont)"
2580 PRINT
2590 PRINT "TRS-80 CROMEMCO"
2600 PRINT
2610 PRINT "(filename)/(extension) (filename).(extension)"
2620 PRINT
2630 PRINT "In the two examples, (filename) is an alphabetical character"
2640 PRINT "string no longer than eight (8) characters. (extension) is "
2650 PRINT "a file extension name that is also an alphabetical character"
2660 PRINT "string. The extension must not be longer than three (3)"
2670 PRINT "characters. The extension is OPTIONAL but, if used, must"
2680 PRINT "follow the format EXACTLY. NUMBERS may be used in both "
2690 PRINT "filenames and extensions, but they must NOT be the FIRST letter"
2700 PRINT
2710 INPUT "press ENTER":T$
2720 GOSUB 8120
2730 PRINT "Filenames (cont)"
2740 PRINT "Here are some examples of legal filenames for your computer"
2750 PRINT
2760 PRINT "MYPROG/BAS MYPROG.BAS"
2770 PRINT "MRHAPPY/BAS MRHAPPY.BAS"
2780 PRINT "SWIMFIN2/BAS SWIMFIN"
**** 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
2920 PRINT* they mean something to the programmer who made them. It
2930 PRINT* would be unwise to name your program something common,
2940 PRINT* like TEST.BAS, because someone else has probably already used
2950 PRINT* that name. If they have, you will destroy their program when
2960 PRINT* you SAVE your program to disk. Each program name must be
2970 PRINT* unique.
2980 INPUT* press ENTER: T$
2990 GOSUB B120
3000 PRINT* Filenames (cont)
3010 PRINT
3030 PRINT* SWIMFIN.BAS
3040 PRINT
3040 PRINT* Notice, in the above filename, the extension is BAS. This
3050 PRINT* would normally indicate that the file is a BASIC file (you:
3060 PRINT* may have word processor files, machine language files, or
3070 PRINT* a variety of others). A good tip is to always save your
3080 PRINT* BASIC files with this extension. That way, when you read
3090 PRINT* the disk directory, you can tell that you have to go to BASIC
3100 PRINT* to run any program that has the extension - .BAS.
3110 PRINT
3120 INPUT* 'press ENTER: T$
3130 GOSUB B120
3140 PRINT* Filenames (cont)
3150 PRINT
3160 PRINT* This section is very important to you. It has shown you what
3170 PRINT* filenamess are and what legal filenames look like. In the
3180 PRINT* future you will use them a lot. Be sure you under-
3190 PRINT* stand the idea behind filenames before you continue. It
3200 PRINT* would be a good idea to look them up in your operating
3210 PRINT* manual for Microsoft BASIC. There is many rules that were not
3220 PRINT* covered here, but the rules we covered will get you by for now.
3230 PRINT
3240 INPUT* 'press ENTER: T$
3250 GOSUB B120
3260 INPUT* 'ENTER a 'T' if using a RS80. or 'C' if CPM or CP/BASIC 4.0:
3270 IF T$ = "T" GO TO B5A0
3280 GOSUB B120
3290 PRINT* In the Commodore systems, which of the following would be
3300 PRINT* considered a legal filename:
3310 PRINT
3320 PRINT* T1111111FIL
3330 PRINT* T lessons.243
3340 PRINT* TEST.BAS
**** Listing of Program 'LESSON2' ****

2250 PRINT" LUNCHTALK"
2260 PRINT
2270 INPUT"press the letter opposite the correct answer and press ENTER":T$
2280 IF T$ = "A" goto 3250
2290 PRINT
2300 PRINT"WRONG the correct answer is A (XXXXX.XXX.FIL)"
2310 PRINT
2320 GOTO 3250
2330 PRINT"CORRECT - good job"
2340 PRINT
2350 INPUT"press ENTER":T$
2360 GOSUB 9120
2370 PRINT"Is the extension necessary for a filename to be legal?"$
2380 PRINT
2390 PRINT"YES"$
2400 PRINT"NO"
2410 PRINT
2420 INPUT"press the letter opposite the correct answer and press ENTER":T$
2430 PRINT
2440 IF T$ = "B" goto 3470
2450 PRINT"WRONG - the correct answer is B (the extension is not needed)"
2460 GOTO 3470
2470 PRINT"CORRECT"
2480 PRINT
2490 INPUT"press ENTER":T$
2500 GOSUB 9120
2510 GOSUB 970
2520 IF T$ = "S" goto 2530
2530 RETURN
2540 GOSUB 9120
2550 PRINT"Which of the following filenames is correct?"
2560 PRINT
2570 PRINT"A  XXXXXXX.BAS"
2580 PRINT"B  24:lesson.BAS"
2590 PRINT"C  TEST.BAS"
2600 PRINT"D  LUNCHTALK"
2610 PRINT
2620 INPUT"press the letter opposite the correct answer and press ENTER":T$
2630 PRINT
2640 IF T$ = "A" goto 2670
2650 PRINT"WRONG the correct answer is A (XXXXX.XXX.BAS)"
2660 GOTO 2670
2670 PRINT"CORRECT"
2680 PRINT
2690 INPUT"press ENTER":T$
2700 GOSUB 9120.
**** Listing of Program 'LESSON2' ****

7710 PRINT 'An extension to a filename is mandatory.'
7720 PRINT
7730 PRINT 'YES'
7740 PRINT 'NO'
7750 PRINT
7760 INPUT 'Press the letter opposite the correct answer and press ENTER': T$.
7770 PRINT
7780 IF T$ = 'S' GOTO 7810
7790 PRINT 'WRONG - the correct answer is B (an extension is not needed).'
7800 GOTO 7820
7810 PRINT 'CORRECT'
7820 PRINT
7830 INPUT 'Press ENTER': T$.
7840 GOSUB 9270
7850 IF T$ = 'S' GOTO 7870
7860 RETURN
7870 GOSUB 9120
7880 PRINT 'SAVE, LOAD and RUN'
7890 PRINT
7900 PRINT
7910 PRINT 'At the beginning of the last section, we said we would discover'
7920 PRINT 'how to SAVE our program so we wouldn’t have to keep typing'
7930 PRINT 'them in all the time. Well, this is it. To SAVE your program,'
7940 PRINT 'let’s say you called it MYPROG.BAS. all you do is:'
7950 PRINT
7960 PRINT '1) Type in the program'
7970 PRINT '2) Type SAVE "MYPROG.BAS"; (cont.)'
7980 PRINT '3) Congratulations on a good job!
7990 PRINT
8000 PRINT 'Be sure to notice that the filename is enclosed in quotation marks. If you don’t enclose the name in quotation marks, the command will "FAIL" because it will fail.'
8010 PRINT
8020 INPUT 'Press ENTER': T$.
8030 GOSUB 9120
8040 PRINT 'SAVE, LOAD, and Run (cont.)'
8050 PRINT
8060 PRINT 'There are other things you should be aware of before you try'
8070 PRINT 'to SAVE a program. First, there should be enough room on the'
8080 PRINT 'disk to hold the program. If you are using a Commodore disk'
8090 PRINT 'you will probably not have any problem in this area. Ask'
8100 PRINT 'your system operator for more information. If you are using a 80KB or 80KB-90K floppy disk, then go to the COMMAND mode by:
8110 PRINT 'using the CMD<CHR$ (84)<<CHR$ (84)<<CHR$ (84) and then typing in DIR or'
8120 PRINT 'DIR II. Drive 1 is called 10 and drive 2 is called 11. Then'
8130 PRINT 'watch the display, you will see the free space left on the disk.'
8140 PRINT 'and a DIRECTION of the files on the disk. If you have over'}
1. Print "To begin, you have enough room for almost any program.
2. Print "See your operating manual for more details."
3. Input "Press enter."
4. Gobosub 31
5. Print "SAVE, RUN, and LOAD (cont.)"
6. Print "Remember, if you have enough space, then just type in this:
7. Print "SAVE" CHR$(84) MYPROG.BAS CHR$(84) or "SAVE"CHR$(84)MYPROG.BASCHR$(84)
8. Print "if using a TRS-80"
9. Print "Now that we know how to SAVE a program, how do we get it back?"
10. Print "So we can RUN it again. That’s easy. Just type in:"
11. Print "RUN"CHR$(84)MYPROG.BASCHR$(84)
12. Print "Press enter."
13. Gobosub 31
14. Print "SAVE, RUN, and LOAD (cont.)"
15. Print "RUN"CHR$(84)MYPROG.BASCHR$(84)
16. Print "In the command, the computer will load your"
17. Print "program that you had previously saved, and run it."
18. Print "What if you just saved a piece of a program because you were
19. Print "Forgot, expecting to come back at a later date and add to it?"
20. Print "If you did that, then you wouldn’t want to RUN the program.
21. Print "You would just want to LOAD the program and LIST it to be"
22. Print "sure it was the right one, then add the line numbers you need."
23. Print "Let’s complete the program."
24. Print "Press enter."
25. Gobosub 31
26. Print "SAVE, RUN, and LOAD (cont.)"
27. Print "You would LOAD the program using the same format as for SAVING:
28. Print "and RUNNING it. That is:"n
29. Print "LOAD"CHR$(84)MYPROG.BASCHR$(84)
30. Print "Remember sure to save the program again after you modify it, because"
31. Print "only a copy of the old version will be on the disk?"
32. Print "Press enter."
33. Gobosub 31
4320 GOSUB 8120
4330 PRINT "Which of the following is the correct command to SAVE a"  
4340 PRINT "program named FRITI?"
4350 LET T$ = "A"
4360 GOSUB 4770
4370 PRINT "B" : GOTO 4770
4380 PRINT "C" : GOTO 4770
4390 PRINT "D" : PRINT "SAVED"  
4390 PRINT "SAVED"
4400 PRINT  
4410 INPUT "Press the letter opposite the correct answer and press ENTER": T$  
4420 IF T$ = "A" THEN 4770
4430 IF T$ = "B" THEN 4770
4440 IF T$ = "C" THEN 4770
4450 IF T$ = "D" THEN 4770
4460 PRINT "You have just gotten to BASIC and want to load a program you"  
4470 PRINT "have been working on. The filename for the program is SQUEEZE;"
4480 PRINT "How would you get the program from permanent storage to "  
4490 PRINT " temporary memory? Which of the following would you type in?"
4500 PRINT  
4510 PRINT  
4520 PRINT "LOAD CHR$(27)" : SQUEEZE:CHR$(27)"
4530 PRINT "LOAD CHR$(27)" : SQUEEZE.BAS:CHR$(27)"
4540 PRINT  
4550 INPUT "Press the letter opposite the correct answer and press ENTER": T$  
4560 IF T$ = "A" THEN 4770
4570 IF T$ = "B" THEN 4770
4580 IF T$ = "C" THEN 4770
4590 IF T$ = "D" THEN 4770
4600 PRINT "You should start early in our programming life, isn"  
4610 PRINT "t"  
4620 PRINT "DOCUMENTATION of how a program runs. You can include state-"  
4630 PRINT "statements within a program THAT WILL NOT BE TOUCHED BY THE COMPUTER-
4640 PRINT "and will add to the clarity of your program. That way, when"
4650 PRINT "YOU VIEW YOUR PROGRAM (OR WHEN ANOTHER PROGRAMMER DOES), YOU"
4660 PRINT "can read the reminder left behind and more fully understand"
50 REM print "The program."
51 INPUT "Which statements are called REM statements (REMARK statements)."
52 PRINT
53 INPUT "Press ENTER":T$  
54 GOSUB 8120
55 PRINT REM (cont)
56 PRINT
57 PRINT "The format for a REM statement is LINE # REM: REMARK."  
58 PRINT
59 PRINT "An example is:" 
60 PRINT
61 PRINT "This is now a REMARK statement is made:"  
62 PRINT
63 PRINT "When the computer sees REM it ignores all data that follows."  
64 PRINT
65 INPUT "Press ENTER":T$  
66 GOSUB 8120
67 PRINT "Here is an example of REMarks in a program:"  
68 PRINT
69 PRINT "The variable in the next line is approx equal to PI:"  
70 PRINT
71 PRINT "C = 1+2*2:"  
72 PRINT
73 PRINT "PRINT C:"  
74 PRINT
75 PRINT "Which would give:"  
76 PRINT
77 PRINT "12.65"  
78 PRINT
79 PRINT "Notice that the REM statement was not printed"  
80 PRINT
81 INPUT "Press ENTER":T$  
82 GOSUB 9120
83 PRINT "CHR$(14): "SWEETUMS is my girl."CHR$(14):" 
84 PRINT "CHR$(14): "SWEETUMS is my girl."CHR$(14):" 
85 PRINT "CHR$(14): "Her real name is CANDY."CHR$(14):" 
86 PRINT "CHR$(14): "I love her."CHR$(14):" 
87 PRINT "RUN:"  
88 PRINT
89 PRINT "The above program would print the following: TRUE OR FALSE:"  
90 PRINT
91 PRINT "SWEETUMS is my girl." "Her real name is CANDY:"  
92 PRINT
93}
Listing of Program "LESSON2" 07/10/83 - 00:01:13

#### Listing of Program 'LESSON2' ####

```
5550 INPUT "True in TRUE or FALSE, whichever is correct":T$
5560 PRINT
5570 IF T$ = "TRUE" GOTO 5610
5580 PRINT "WARNING - the correct answer is TRUE - REM statements are not"
5590 PRINT "printed!"
5600 GOTO 5620
5610 PRINT "CORRECT - good job!"
5620 PRINT
5630 INPUT "press ENTER":T$
5640 GOSUB 3270
5650 IF T$ = "B" GOTO 5900
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 = 10"
5740 PRINT "PRINT 10"
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 CHRS(24) "press the correct letter, then press ENTER" CHRS(24)"
5950 PRINT
5960 INPUT "press ENTER":T$
5970 GOSUB 3120
5980 PRINT "INPUT Statements (cont)"
5990 PRINT
6000 PRINT "INPUT CHRS(24) "press the correct letter, then press ENTER" CHRS(24)"
```

96
### Listing of Program 'LESSON2' ###

```plaintext
010 PRINT*RUN
020 PRINT
030 PRINT"Gives us:"  
040 PRINT
050 PRINT"press the correct letter, then press ENTER":  
060 PRINT
070 PRINT"Notice that a question mark is automatically inserted after the"
080 PRINT"message is printed. When the question is answered, the letter"
090 PRINT"that the student selects is assigned to TS, just as if we"
100 PRINT"had assigned a value to it in an equals statement."
110 PRINT"Also note that a semi-colon is placed after the text":  
120 PRINT
130 INPUT"press ENTER";TS
140 GDUB 8120
150 PRINT"Gives us:"  
160 PRINT
170 PRINT"Here is another example:"  
180 PRINT
190 PRINT"A = 20"
200 PRINT*20 INPUT"CHR$(14);Enter a number between 1 and 9";IN
210 PRINT"C = AN"  
220 PRINT*RUN
230 PRINT
240 PRINT"Gives us:"  
250 PRINT"Enter a number between 1 and 9":  
260 PRINT"if we ENTER a 5 then"
270 PRINT
280 PRINT"10"
290 PRINT
300 INPUT"press ENTER";TS
310 GDUB 8120
320 PRINT"Gives us:"  
330 PRINT
340 PRINT*10 INPUT"CHR$(14);Enter a number between 1 and 9";IN
350 PRINT
360 PRINT"We have learned then, that the INPUT statement allows you"
370 PRINT"to ENTER data in a program while it is running. It does"
380 PRINT"this by stopping the program and waiting for you to enter"
390 PRINT"data. When you do, it sets the data equal to the variable"
400 PRINT"on the end of the INPUT statement."
410 PRINT
420 PRINT"Between the message or prompt and the variable,"  
430 PRINT"do not place a semi-colon (look at example above)."
440 PRINT
450 INPUT"press ENTER";TS
460 GDUB 8120
```

07
**Listing of Program 'LESSON2'**

6470 PRINT*  \(\text{INPUT Statements}\)
6480 PRINT
6490 PRINT"You may use the INPUT statement without using a prompt or text"  
6500 PRINT"message. If you do, then you must NOT put in a semi-colon."
6510 PRINT"For example:"  
6520 PRINT
6530 PRINT*10 PRINT*CHR$(34)"When you see a question mark, ENTER a 5"CHR$(34)
6540 PRINT*20 INPUT N*  
6550 PRINT*RUN*  
6560 PRINT  
6570 PRINT"Gives you:"  
6580 PRINT
6590 PRINT"When you see a question mark, ENTER a 5"  
6600 PRINT"**"  
6610 PRINT
6620 INPUT press ENTER*116  
6630 GOSUB B10  
6640 PRINT*  \(\text{INPUT Statements (cont)}\)*:PRINT
6650 PRINT*10 PRINT*CHR$(34)"When you see a question mark, ENTER a 5"CHR$(34)
6660 PRINT*20 INPUT N*  
6670 PRINT*RUN*  
6680 PRINT
6690 PRINT"When you see a question mark, ENTER a 5"  
6700 PRINT***  
6710 PRINT
6720 PRINT"Notice how the INPUT statement prompt (question mark) is on"  
6730 PRINT"the following line? If we hadn't included line 10, we wouldn't"  
6740 PRINT"know what to do when we saw the question mark. That's why you"  
6750 PRINT"will see the text included in an INPUT statement most of the"  
6760 PRINT"time. However, both ways are used."
6770 PRINT
6780 INPUT press ENTER*116  
6790 GOSUB B10  
6800 PRINT*  \(\text{INPUT Statement (cont)}\)*:PRINT
6810 PRINT
6820 PRINT*10 INPUT*CHR$(34)"ENTER a number between 1 and 9"CHR$(34)":"  
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."

---

98
**Listing of Program 'LESSON2'**

`100 PRINT
110 INPUT"press ENTER":T$`

`120 PRINT INPUT Statements (cont)"`

`130 PRINT
140 PRINT "We have seen examples of both STRING variables and NUMERIC"
150 PRINT "variables and we have seen both used with INPUT statements."
160 PRINT
170 PRINT "If you try to ENTER string data into a NUMERIC variable, you"
180 PRINT "will get a "REDO" message. That means you tried to ENTER data"
190 PRINT "that was not proper for a NUMERIC variable."
200 PRINT
210 PRINT "A problem develops when you think you are entering NUMERIC data"
220 PRINT "and you ENTER it into a STRING. You will not get an error"
230 PRINT "message. Remember, put the right kind of variable on the end"
240 PRINT "of the INPUT statement (you will be tested on this)."
250 PRINT
260 INPUT"press ENTER":T$
270 GOSUB B120`

`280 PRINT
290 PRINT INPUT Statements (cont)"`

`300 PRINT
310 PRINT "You may ENTER data into more than one variable using only ONE"
320 PRINT "INPUT statement. Just put a comma ',' between the variables."
330 PRINT "and a question mark will be prompted for each variable."
340 PRINT "For example:"`
350 PRINT
360 PRINT "100 INPUT"CHR$(144)"ENTER three (3) numbers"CHR$(144)"A,B,C"
370 PRINT "RUN"
380 PRINT
390 PRINT "ENTER three (3) numbers? 10"
400 PRINT "*** 22"
410 PRINT "*** 5"
420 PRINT
430 PRINT "230 INPUT"press ENTER":T$
440 GOSUB B120`

`450 PRINT
460 PRINT INPUT Statements (cont)"`

`470 PRINT
480 PRINT "100 INPUT"CHR$(144)"ENTER three (3) numbers"CHR$(144)"A,B,C"
490 PRINT "RUN"
500 PRINT
510 PRINT "ENTER three (3) numbers? 10"
520 PRINT "*** 22"
530 PRINT "*** 5"
540 PRINT
550 PRINT "Notice how the computer keeps prompting you with question marks"
560 PRINT "until it gets all of its data. This can be a very useful"`
**** 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 INPUT"press ENTER":T$
7480 GOSUB 8120
7490 PRINT" ENTER three (3) numbers: 10,22.5"
7500 PRINT
7510 PRINT" Notice that we hastened our data input, in this case, by"
7520 PRINT" entering all the data on one line, separated by commas."
7530 PRINT
7540 PRINT" You may choose either way of entering the data, it makes no"
7550 PRINT" difference."
7560 PRINT
7570 PRINT
7580 INPUT"press ENTER":T$
7590 GOSUB 8120
7600 PRINT" An INPUT statement CAN assign a value to a variable while the"
7610 PRINT" program is running? (TRUE or FALSE)"
7620 PRINT
7630 INPUT" ENTER the word TRUE or ENTER the word FALSE":T$
7640 PRINT
7650 IF T$ = "TRUE" GOTO 7690
7660 PRINT" WRONG - an INPUT statement IS used for inputting data while"
7670 PRINT" the program is running"
7680 GOTO 7700
7690 PRINT" CORRECT"
7700 PRINT
7710 INPUT"press ENTER":T$
7720 GOSUB 8120
7730 PRINT" What is the prompt that an INPUT statement ALWAYS gives?"
7740 PRINT
7750 PRINT" A A question mark"
7760 PRINT" B Two question marks"
7770 PRINT" C Quotes"
7780 PRINT" D the word INPUT"
7790 PRINT
7800 INPUT" ENTER the correct letter (either A,B,C, or D)":T$
7810 PRINT
7820 IF T$ = "A" GOTO 7850
7830 PRINT" WRONG - the prompt that is ALWAYS given is a question mark"
7840 GOTO 7860

160
7950 PRINT "CORRECT - GREAT!"
7960 PRINT
7970 INPUT "press ENTER": T$
7980 GOSUB 8120
7990 PRINT "Which of the following is a valid response to this statement:"
8000 PRINT
8010 PRINT *20 INPUT A,B,C$
8020 PRINT
8030 PRINT A. 2.10.15"
8040 PRINT B. 2.10.15 North Ela"
8050 PRINT C. TWO.1.22"
8060 PRINT D. 22.30.NONE,E"
8070 PRINT
8080 INPUT ENTER the letter opposite the correct response and press ENTER": T$
8090 PRINT
8100 IF T$ = "B" GOTO 8030
8110 PRINT "WRONG - the correct answer is B (2.10.15 North Elma"
8120 GOTO 8040
8130 PRINT "CORRECT - good, this section is just about done"
8140 PRINT
8150 INPUT "press ENTER": T$
8160 GOSUB 8270
8170 IF T$ = "A" GOTO 8570
8180 RETURN
8190 REM **
8200 REM ** This subroutine clears the screen on any terminal
8210 REM **
8220 FOR X = 1 TO 24
8230 PRINT
8240 NEXT X
8250 RETURN
8260 PRINT *LESSON 2"
8270 PRINT
8280 PRINT "This is the first part of a two part lesson"
8290 PRINT "It is divided into the following sections."
8300 PRINT
8310 PRINT 1) Introduction  4) Remaks"
8320 PRINT 2) F. Lennes  5) Input Statements"
8330 PRINT 3) SAVE, LOAD, RUN"
8340 PRINT
8350 PRINT
8360 PRINT
8370 REM
8380 RETURN
8390 GOSUB 8120
8400 PRINT "Which do you wish to do?"
8410 PRINT
8420 PRINT
8430 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 8370
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$ = "A" GOTO 7690
350 IF T$ = "C" GOTO 7700
360 IF T$ = "B" GOTO 430
370 IF T$ = "A" GOTO 280
380 GOSUB 550
390 GOSUB 5900
400 GOSUB 4720
410 GOSUB 6000
420 GOSUB 6640
430 GOSUB 6950
440 GOSUB 6990
450 PRINT
460 PRINT'Please type in the number beside the area you wish"
470 PRINT'to review (1 through 5) and then press ENTER - press 0 and"
480 PRINT'press ENTER to return to the Menu.""
490 PRINT
500 INPUT'What is our choice';N
510 IF N = 0 THEN GOTO 7690
520 IF N = 5 THEN GOTO 762
530 CN N GOSUB 550,5900,4720,6000,6640
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
LISTING OF PROGRAM 'LESSON2A'

720 PRINT 'READ and DATA (cont)'
730 PRINT
740 PRINT 'READ and DATA statements are used like the equals sign "=" is'
750 PRINT 'used, but they are much faster and more versatile. Also,'
760 PRINT 'READ and DATA statements OPERATE WITHIN THE PROGRAM INSTEAD'
770 PRINT 'OF INTERFACING YOU WITH THE PROGRAM.'
780 PRINT
790 PRINT 'READ and DATA are two separate statements, but they are ALWAYS'
800 PRINT 'used with each other. The READ statement assigns the data'
810 PRINT 'as the program runs, and the DATA statement holds the values'
820 PRINT 'to be assigned.'
830 PRINT
840 INPUT 'press ENTER':T$
850 GOSUB 6950
860 PRINT 'READ and DATA (cont)'
870 PRINT
880 PRINT 'The format of the READ statement is: '
890 PRINT
900 PRINT '(line number) READ (variable or variables)'
910 PRINT
920 PRINT 'An example of a READ statement that would read values into '
930 PRINT 'variables Bl, A, and XX$ is:'
940 PRINT
950 PRINT 'READ Bl, A, XX$'
960 PRINT
970 INPUT 'press ENTER':T$
980 GOSUB 6950
990 PRINT 'READ and DATA (cont)'
1000 PRINT
1010 PRINT '10 READ Bl, A, XX$'
1020 PRINT
1030 PRINT 'Note that both numeric and string variables may be 'read' '
1040 PRINT 'An example of a DATA statement that would be read is:'
1050 PRINT
1060 PRINT 'DATA 22,15, "CHR$(34)"AIN'T she sweet?"CHR$(34)"
1070 PRINT
1080 PRINT 'Notice that the DATA statement has a different line number, but '
1090 PRINT 'it follows the same format as the READ statement. When these'
1100 PRINT 'two statements are placed in a program, the variables Bl, A, '
1110 PRINT 'and XX$ would hold 22, 15, and "CHR$(34)"AIN'T she sweet?"CHR$(34)" respectively.'
1120 PRINT
1130 INPUT 'press ENTER':T$
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:"
1270 GO TO 1280
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:"
1350 PRINT
1360 IF TS = "TRUE" THEN GOTO 1390
1370 PRINT "Wrong - the sentence is valid."
1380 GO TO 1390
1390 PRINT "Correct."
1400 PRINT
1410 INPUT "Press ENTER:"
1420 GO TO 1430
1430 PRINT "READ and DATA (cont)"
1440 PRINT
1450 PRINT "READ B1,A,Y"
1460 PRINT
1470 PRINT "DATA 22.5,CHR$(4):AINT she sweet:"CHR$(4)"
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:"
1570 GO TO 1580
1580 PRINT "READ and DATA (cont)"
1590 PRINT "READ B1,A,Y"
1600 PRINT "PRINT B1,A,Y"
1610 PRINT "PRINT CHR$(4)" "The average of these numbers is:"CHR$(4)"
1620 PRINT "PRINT (B1+A+Y)/3"
1630 PRINT "DATA 5.10.15"
1640 PRINT "RUN"
### Listing of Program 'LESSONZA' ####

```
1650 PRINT
1660 PRINT "Gives us:"
1670 PRINT
1680 PRINT 5 10 15
1690 PRINT "The average of these numbers is 10"
1700 PRINT
1710 PRINT "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 READ and DATA (cont)"
1770 PRINT
1780 PRINT "The calculations were not printed on another line because of"
1790 PRINT "the semi-colon after the print statement in line 30."
1800 PRINT
1810 PRINT 10 READ B1.A,Y
1820 PRINT 20 PRINT B1.A,Y
1830 PRINT 50 PRINT "CHR$(24):"The average of these numbers is"CHR$(24)":""
1840 PRINT 40 PRINT B1.A,Y
1850 PRINT 50 DATA 5,10,15
1860 PRINT
1870 PRINT "Line 10 read the data in line 50, line 20 PRINTed it, and"
1880 PRINT "line 30 printed the message. The calculations in line"
1890 PRINT "were printed on the end of the message due to the semi-colon"
1900 PRINT "at the tail of line 20."
1910 PRINT
1920 INPUT "press ENTER": T$
1925 GOSUB 6950
1940 PRINT READ and DATA cont)"
1950 PRINT
1960 PRINT "Let's look at it once more:"
1970 PRINT
1980 PRINT 10 READ B1.A,Y
1990 PRINT 20 PRINT B1.A,Y
2000 PRINT 50 PRINT "CHR$(24):"The average of these numbers is"CHR$(24)":""
2010 PRINT 40 PRINT B1.A,Y:15
2020 PRINT 50 DATA 5,10,15
2030 PRINT
2040 PRINT "Save us:"
2050 PRINT
2060 PRINT 5 10 15
2070 PRINT "The average of these numbers is 10"
2080 PRINT
2090 INPUT "press ENTER": T$
2100 GOSUB 6950
```

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

2110 PRINT "Now for a neat example:"
2120 PRINT
2130 PRINT "10 READ A1,B1,C1"
2140 PRINT "10 PRINT A1,B1,C1"
2150 PRINT "10 DATA 1,2"
2160 PRINT "RUN"
2170 PRINT
2180 PRINT "Gives us:"
2190 PRINT
2200 PRINT "OUT OF DATA IN 10"
2210 PRINT
2220 PRINT "The BASIC language processor gave us an ERROR message that"
2230 PRINT "indicates we didn't have enough data for the number of variables that we tried to READ."
2240 PRINT
2250 PRINT
2260 INPUT "press ENTER:"
2270 GOSUB 8750
2280 PRINT "READ and DATA (cont)"
2290 PRINT
2300 PRINT "10 READ A1,B1,C1"
2310 PRINT "20 PRINT A1,B1,C1"
2320 PRINT "30 DATA 1,2"
2330 PRINT
2340 PRINT "This program will bomb (fail) because it will try to find a 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."
2350 PRINT
2360 INPUT "press ENTER:"
2370 GOSUB 8750
2380 PRINT "READ and DATA (cont)"
2390 PRINT
2400 PRINT "10 READ A1,B1"
2410 PRINT "20 PRINT A1,B1"
2420 PRINT "30 DATA 1,2,3,4,99"
2430 PRINT
2440 PRINT "Gives us:"
2450 PRINT
2460 PRINT "10 READ A1,2"
2470 PRINT "30 PRINT A1,2"
2480 PRINT
2490 PRINT "Although there were more data points that could have been read,"
2500 PRINT "there were no more variables left to READ them, so the program stopped. This would not cause an ERROR message."
2510 PRINT
2520 INPUT "press ENTER:"

107
**Listing of Program 'LESSONZA'**

```
2570 GOSUB 6950
2580 PRINT "Is the following program valid?"
2590 PRINT
2600 PRINT "10 READ X,Y,Z"
2610 PRINT "20 DATA 25,2,15,35,64"
2620 PRINT "30 PRINT I"
2630 PRINT
2640 PRINT "A Yes"
2650 PRINT "B No"
2660 PRINT
2670 INPUT "Enter the letter opposite the correct answer:";
2680 PRINT
2690 IF T$ = "A" THEN GOTO 220
2700 PRINT "WRONG - the program is valid"
2710 GOTO 2730
2720 PRINT "CORRECT";
2730 PRINT
2740 INPUT "press ENTER:";
2750 PRINT
2760 INPUT "What is the value of I that will be printed out:";
2770 PRINT
2780 IF T$ = "5" THEN GOTO 2830
2790 PRINT "WRONG - the correct answer is 5. I is the third variable to be read. so the third data point is cut out in it."
2800 PRINT "CORRECT";
2810 GOTO 2840
2820 PRINT "CORRECT";
2830 GOSUB 6750
2840 PRINT
2850 INPUT "press ENTER:";
2860 GOSUB 6950
2870 PRINT "READ and DATA (cont."
2880 PRINT
2890 PRINT "Suppose you want to READ the same data points into DIFFERENT variables. Or perhaps you are making a program that will deal with 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."
2900 PRINT
2910 GOSUB 6950
2920 PRINT "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."
2930 PRINT
2940 PRINT 3020 PRINT "10 READ A1,B1"
```
***** Listing of Program 'LESSON2A' ***** 07/10/83 - 00:24:03

3030 PRINT"20 PRINT A1.B1"
3040 PRINT"30 RESTORE"
3050 PRINT"40 READ C1.D1"
3060 PRINT"50 PRINT C1.D1"
3070 PRINT"60 DATA 1.2"
3080 PRINT"RUN"
3090 PRINT
3100 PRINT"1*
3110 PRINT"1*
3120 PRINT
3130 INPUT"Study this example closely and then press ENTER":T$
3140 GOSUB 6950
3150 PRINT READ and DATA (cont)
3160 PRINT
3170 PRINT"This has been an extra long section and, if you are new to"
3180 PRINT"BASIC, you probably are confused about some of the rules,"
3190 PRINT"Don't let that worry you. Get out your BASIC manual (it came"
3200 PRINT"with your computer), and, after the little quiz that is"
3210 PRINT"coming up, go back and review this section again. THEN"
3220 PRINT"PRACTICE the techniques once you are through with this"
3230 PRINT"lesson. It is very important that you start practicing what"
3240 PRINT"you are learning. Practice with the manual beside you, and"
3250 PRINT"don't be afraid to ask an experienced programmer when you are"
3260 PRINT"confused. Use the examples seen in this program, or make up"
3270 PRINT"your own. You are one step closer to being a BASIC programmer!"
3280 PRINT
3290 INPUT"Press ENTER":T$
3300 GOSUB 6950
3310 PRINT"Is the following statement TRUE or FALSE?"
3320 PRINT
3330 PRINT"The RESTORE statement causes the READ:DATA combination"
3340 PRINT"to reset to the first data point to the right of the"
3350 PRINT"first DATA statement.""
3360 PRINT
3370 PRINT"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$

109
#### Listing of Program 'LESSON2A' ####

7490 GOSUB 6950
7500 PRINT 'READ and DATA statements are used within a program, and they'
7510 PRINT 'do NOT stop the program so DATA can be entered.'
7520 PRINT
7530 PRINT 'A TRUE'
7540 PRINT 'B FALSE'
7550 PRINT
7560 INPUT *press the letter opposite the correct answer and press ENTER' '
7570 PRINT
7580 IF T = 'A' THEN GOTO 3520
7590 PRINT 'WRONG - READ and DATA statements DO NOT halt the program, they'
7600 PRINT 'must be used within the program.'
7610 GOTO 3520
3520 PRINT 'CORRECT - you certainly have a good memory.'
7630 PRINT
7640 INPUT *press ENTER to'
7650 GOSUB 6950
7660 PRINT 'D READ 41.51'
7670 PRINT 'D PRINT 41.51'
7680 PRINT 'D RESTORE'
7690 PRINT 'D READ 1.31,1.61'
7700 PRINT 'D PRINT 1.31,1.61'
7710 PRINT 'D DATA 342,34'
7720 PRINT
7730 PRINT 'The above program is ERROR free'
7740 PRINT
7750 PRINT 'A TRUE'
7760 PRINT 'B FALSE'
7770 PRINT
7780 INPUT *enter the correct answer (either A or B)'
7790 PRINT
7800 IF 'T = 'G' GOTO 3840
7810 PRINT 'WRONG - LINE 4 tried to read more DATA than was available.'
7820 PRINT 'even though the RESTORE command was used.'
7830 GOTO 3520
3840 PRINT 'CORRECT'
7850 PRINT
7860 INPUT *press ENTER to'
7870 GOSUB 1530
3880 IF T = 'G' THEN GOSUB 550
3890 RETURN
7900 GOSUB 6950
7910 GOSUB 720
7920 PRINT 'There are two types of branches, and we will be studying them'
7930 PRINT 'in the next two sections. They are CONDITIONAL branches, and'
7940 PRINT 'INCONDITIONAL branches. The above program has both kinds in it.'
**Listing of Program 'LESSON4A'**

```
07/10/83 - 00:24:13

7250 PRINT
7260 PRINT'Line 30 is CONDITIONAL and line 40 is UNCONDITIONAL. Can you'
7270 INPUT'see why? Study this for a moment and then press ENTER'IT
7280 GOSUB 6590
7290 GOSUB 7220
7300 PRINT'Line 30 is CONDITIONAL because it will only GOTO line 50 if'
7310 PRINT'the CONDITION that A = 5 is satisfied. That is, control will'
7320 PRINT'only be transferred to line 50 if A = 5.'
7330 PRINT
7340 PRINT
7350 INPUT'press ENTER'IT
7360 GOSUB 6590
7370 GOSUB 7220
7380 PRINT'Line 40 is UNCONDITIONAL because it will ALWAYS GOTO line 20'
7390 PRINT'when it is executed. There will be no choice made.'
7400 PRINT'Control will go to line 20.'
7410 PRINT
7420 PRINT
7430 INPUT'press ENTER'IT
7440 GOSUB 6590
7450 GOSUB 7220
7460 PRINT'Notice that A will not equal 5 until line 20 is executed.'
7470 PRINT'Then, therefore, until A = 5, the CONDITION in line 30 will'
7480 PRINT'not be set and control will NOT GOTO line 50. Instead, it'
7490 PRINT'will go to the next line which is UNCONDITIONAL GOTO line 20.'
7500 PRINT
7510 PRINT
7520 INPUT'Study this carefully, and then press ENTER'IT
7530 GOSUB 6590
7540 GOSUB 7220
7550 PRINT'The statement in line 20 has made a COUNTER out of the variable'
7560 PRINT'. Everytime the line is executed, A is incremented by one.'
7570 PRINT'COUNTERS are very useful in BASIC and we will discuss them'
7580 PRINT'here in a future lesson. For now, try to understand how this'
7590 PRINT'type of loop works, and it will help you immensely in the future.'
7600 PRINT
7610 PRINT
7620 PRINT'press ENTER'IT
7630 GOSUB 6590
7640 PRINT
7650 PRINT'The two types of branching are'
7660 PRINT
7670 PRINT'CONDITIONAL and UNCONDITIONAL'
7680 PRINT'COUNTER and UNCONDITIONAL'
7690 PRINT'IF and GOTO'
7700 PRINT
7710 PRINT
7720 INPUT'the letter opposite the correct answer and press ENTER'IT
7730 PRINT
7740 IF 'A' = 'A' THEN GOTO 440:
7750 PRINT'IF 'B' = 'B' THEN GOTO 440:
7760 PRINT'IF 'C' = 'C' THEN GOTO 440:
7770 PRINT'IF 'D' = 'D' THEN GOTO 440:
7780 PRINT'IF 'E' = 'E' THEN GOTO 440:
```

---

III
**** Listing of Program "LESSON1" ****

440: GOTO 470
440: PRINT "CORRECT"
440: PRINT
440: INPUT "Press ENTERишь"
440: GOSUB 670
440: PRINT "A = "
440: PRINT "What would be the output of this program?"
440: PRINT
440: PRINT "1;" 440: PRINT "2;" 440: PRINT "No output would come from this program"
440: PRINT "1;" 440: PRINT
440: INPUT "Press the letter opposite the correct answer and press ENTER"; 440: GOSUB 670
440: PRINT
440: IF A = 1 THEN GOTO 440
440: PRINT "The correct answer is 1;" 440: GOTO 470
440: PRINT "CORRECT;" 440: PRINT
440: INPUT "Press ENTER"; 440: GOSUB 670
440: IF A = 1 THEN GOTO 470
440: RETURN
440: INPUT "IF Statements;" 440: PRINT
440: PRINT "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.

440: PRINT
440: PRINT "The IF statement causes the program to have decision points. It is one of the most powerful elements of the BASIC language. You have already seen how it can be used in the;" 440: PRINT previous section. In this section, we will explain it in a little more detail.
440: PRINT
440: PRINT "Press ENTER"; 440: GOSUB 670

********
457) BESDE "TD)
458) PRINT "This example shows three variations of the IF statement."
459) PRINT "1) We RUN this program, and ENTER a 'U' when prompted by line"
460) PRINT "10, then the CONDITIONAL statement in line 20 will be satisfied"
461) PRINT "and the program will stop."
462) PRINT
463) PRINT "2) ENTER a number ENTER:"
464) GO TO 100
465) PRINT "IF we ENTER a 10 when prompted by line 10, the CONDITIONAL"
466) PRINT "statement in line 20 will not be satisfied, nothing will happen"
467) PRINT "until control passes to line 30. At that time, the check for"
468) PRINT "If we are greater than 10 will be set, and the message will be printed."
469) PRINT "We get the other condition set to line 50, but the"
470) PRINT "UNCONDITIONAL GOTO at line 50 will send control back to the"
471) PRINT "beginning of the program."
472) PRINT
473) INPUT "Press ENTER to continue:"
474) GO TO 100
475) PRINT "3) IF we ENTER a -10 when prompted by line 10, we will satisfy the"
476) PRINT "CONDITIONAL statement in line 20, the message will be printed"
477) PRINT "and control will eventually get back to line 10."
478) PRINT
479) PRINT "In this program, IF statement (cont.)"
480) PRINT
481) PRINT "The IF statement can also be used to compare two expressions."
482) PRINT "such as:"n
483) PRINT "PRINT "IF 1*2=3 THEN GOTO 19"
484) PRINT "PRINT "Then we assign variable assign can be done in an IF statement:"n
485) PRINT "PRINT "IF 2*2=4 THEN A=4"
486) PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT "PRINT 102}
***** Listing of Program 'LESSON24' *****

5240 PRINT"to line 10; if it does NOT equal 1 then control transfers to '
5250 PRINT"200. In this case, something ALWAYS happens at line 200."
5260 PRINT"because of the ELSE statement."
5270 PRINT
5280 PRINT"200 IF A = 1 THEN GOTO 10 ELSE IF A = 2 GOTO 30"
5290 PRINT
5300 PRINT"In this case, if A = 1 or A = 2 then something will happen."
5310 PRINT"In line 220, if none of the CONDITIONS are met, then the line"
5320 PRINT"will not be executed."
5330 PRINT
5340 INPUT"press ENTER to continue":;
5350 GOTO 9990
5360 PRINT IF Statement (cont)"
5370 PRINT
5380 PRINT"IF (true/false expression) THEN (action) ELSE (action)"
5390 PRINT
5400 PRINT"The IF statement instructs the computer to test the following"  
5410 PRINT"logical or relational expression. If the expression is TRUE"
5420 PRINT"then control will proceed to the action line after the THEN"  
5430 PRINT"word. If the expression is not true, then control will proceed"  
5440 PRINT"to the ELSE action."  
5450 PRINT
5460 INPUT"press ENTER to continue":;
5470 GOTO 9990
5480 PRINT
5490 PRINT"What is the output of the following program?"
5500 PRINT
5510 PRINT"10 A = 255"
5520 PRINT"10 IF A > 190 THEN PRINT "CHR$(14)""TRUE""CHR$(4)"
5530 PRINT"10 ELSE IF A < 254 THEN PRINT "CHR$(34)""TRUE""CHR$(14)"
5540 PRINT"10 PRINT"A = 255"
5550 PRINT
5560 PRINT"10 PRINT"A = 255"
5570 PRINT
5580 PRINT"END"
5590 PRINT
5600 PRINT"A = 255"
5610 PRINT
5620 PRINT
5630 PRINT "ENTER the letter opposite the correct answer":;
5640 PRINT
5650 IF I$ = "C" GOTO 5700
5660 PRINT"WRONG - the correct answer is C (TOO STRONG);"  
5670 PRINT"A = 255;"
5680 GOTO 5700
5690 PRINT"CORRECT - we need "our mind in the budget office!"

114
**** Listing of Program 'LESSON6' ****

570 PRINT
580 INPUT 'Press ENTER to continue'
590 GOSUB 650
592 PRINT 'The IF statement is a CONDITIONAL statement."
592 PRINT
594 PRINT 'Is the above sentence TRUE or FALSE?'
592 PRINT
594 PRINT 'A TRUE'
592 PRINT
594 PRINT 'E FALSE'
600 PRINT
610 INPUT 'Choose the letter opposite the correct answer and press ENTER':T
620 PRINT
691 IF T$ = 'A' GOTO 5940
692 PRINT 'WRONG - the IF statement IS a CONDITIONAL statement."
692 GOTO 5950
694 PRINT 'CORRECT'
695 PRINT
700 INPUT 'Press ENTER to continue'
710 GOSUB 7100
720 IF T$ = 'B' GOTO 4720
730 RETURN
740 GOSUB 3500
750 PRINT"
760 PRINT "GOTO Statements"
770 PRINT
780 PRINT "Conditional branches are written in programs with IF THEN FALSE"
790 PRINT "statements.  Unconditional branches are written with GOTO!"
800 PRINT "statements."
810 PRINT "As we saw earlier, GOTO directs control of a program to another"
820 PRINT "line. For example:"
830 GOSUB 4710
840 PRINT
850 INPUT 'Press ENTER to continue'
860 GOSUB 650
870 GOSUB '470"
880 PRINT
890 PRINT "The GOTO statement in line 40, when executed, sends control to"
900 PRINT "the beginning of the program."
910 PRINT
920 INPUT 'Press ENTER to continue'
930 GOSUB 650
940 PRINT "GOTO (cont)"
950 PRINT
960 PRINT "You can make the GOTO statement a MULTI-way branching statement"
970 PRINT "by modifying it slightly.  For example:"
980 PRINT
990 GOSUB 550

115
***** Listing of Program 'LESSON2A' *****

6260 PRINT
6270 INPUT"press ENTER":T$  
6280 GOSUB 6950  
6290 GOSUB 7580  
6300 PRINT

When line 20 is executed, the value of N is used to count over "
6310 PRINT"N" elements passed the GOTO word. Control branches to the "
6320 PRINT"line number indicated by this Nth' element. If there is no "
6330 PRINT"element that corresponds to the value of N, then control passes "
6340 INPUT"to the next available line. Press ENTER when ready":T$  
6350 GOSUB 6950  
6360 GOSUB 7580  
6370 PRINT

The value of N MUST be greater than 0 and less than 255. If it"  
6390 PRINT"is not. BASIC will print an error. If N is 1, 2, or 3 then the"  
6400 PRINT"program will print the appropriate message and stop."  
6410 PRINT

6420 INPUT"press ENTER":T$  
6430 GOSUB 6950  
6440 PRINT"Which lines (beyond 20) are executed if you ENTER a 10 here?"  
6450 PRINT
6460 GOSUB 7580  
6470 PRINT
6480 PRINT"A 30 and 40"  
6490 PRINT"B 50 and 60"  
6500 PRINT"C 70 and 80"  
6510 PRINT

6520 INPUT"press the letter opposite the correct answer then press ENTER":T$  
6530 PRINT
6540 IF T$ = "A" GOTO 6570  
6550 PRINT"ERROR - the correct answer is A (30 and 40)"  
6560 GOTO 6570  
6570 PRINT"CORRECT"  
6580 PRINT
6590 INPUT"press ENTER":T$  
6600 GOSUB 6950  
6610 PRINT"You are now done with this lesson. When you hit ENTER, your"  
6620 PRINT"will be returned to the MENU where you may review sections"  
6630 PRINT"or take the TEST."  
6640 PRINT
6650 PRINT"Remember, after you are done here, practice some of the"  
6660 PRINT"things you have learned. And keep a BASIC manual by your side."  
6670 PRINT"This program will show you the fundamentals, you have to teach"  
6680 PRINT"yourself how to be good at BASIC. That means YOU must PRACTICE"  
6690 PRINT

110
**** Listing of Program 'LESSON2A' ****

07/10/93 - 00:24:03

6901 INPUT "press ENTER to go to the MENU" : $  
6910 RUN  
6920 REM  **  
6930 REM ** This subroutine clears the screen on any terminal  
6940 REM  **  
6950 FOR X = 1 TO 24  
6960 PRINT  
6970 NEXT X  
6980 RETURN  
6990 PRINT " "  
7000 PRINT  
7100 PRINT "This is the second part of a two part lesson"  
7110 PRINT "It is divided into the following sections:"  
7120 PRINT  
7130 PRINT "1) READ, DATA and RESTORE  3) IF Statements"  
7140 PRINT "2) Branching Introduction  4) GOTO Statements & Summary"  
7150 PRINT " 5) TEST "  
7160 PRINT  
7170 PRINT  
7180 PRINT  
7190 RETURN  
7200 GOSUB 6950  
7210 PRINT "Which do you want to do?"  
7220 PRINT  
7230 PRINT "A. Continue on"  
7240 PRINT "B. Review this lesson again"  
7250 PRINT  
7260 INPUT "press the letter opposite your choice and press ENTER" : $  
7270 IF $ = "A" OR $ = "B" GOTO 7160  
7280 PRINT  
7290 RETURN  
7300 REM  
7310 REM This subroutine prints the Branching section example  
7320 REM  
7330 PRINT "Branching Introduction!"  
7340 PRINT  
7350 PRINT "A = 0"  
7360 PRINT "A = A + 1"  
7370 PRINT "IF A = 5 THEN GOTO 50"  
7380 PRINT "GOTO 20"  
7390 PRINT "PRINT A"  
7400 PRINT "END"  
7410 PRINT  
7420 REM  
7430 RETURN  
7440 REM  
7450 REM This subroutine is for the IF statement examples  
7460 REM  
7470 PRINT "IF Statements (cont)"
***Listing of Program 'LESSON2'***

7360 PRINT
7370 PRINT:10 INPUT "CHR$(34)";"ENTER a number between 1 and 10 to quit:";CHR$(34);"N"
7380 PRINT:20 IF N = 0 THEN STOP
7390 PRINT:30 IF N > 10 THEN PRINT "CHR$(34)";"ERROR - you entered an invalid number;CHR$(34)
7400 PRINT:40 IF N < 0 PRINT "CHR$(34)";"ERROR - you entered an invalid number;CHR$(34)
7410 PRINT:50 GOTO 10
7420 PRINT
7430 RETURN
7440 REM
7450 REM this is subroutine for GOTO example
7460 REM
7470 PRINT
7480 PRINT:10 INPUT "CHR$(34)";"ENTER a number between 1 and 10;CHR$(34);"N"
7490 PRINT:20 IF N = 9 THEN GOTO 50
7500 PRINT:30 IF N > 9 THEN PRINT "CHR$(34)";"GUESS AGAIN;CHR$(34)
7510 PRINT:40 GOTO 10
7520 PRINT:50 PRINT "CHR$(34)";"YOU GUESSED IT"
7530 PRINT:60 END
7540 RETURN
7550 REM
7560 REM this is the subroutine example for ON GOTO
7570 REM
7580 PRINT:10 INPUT "CHR$(34)";"ENTER a number between 1 and 3;CHR$(34);"N"
7590 PRINT:20 ON N GOTO 20,50,70
7600 PRINT:30 PRINT "CHR$(34)";"YOU ENTERED A ONE (or an illegal number);CHR$(34)
7610 PRINT:40 STOP
7620 PRINT:50 PRINT "CHR$(34)";"YOU ENTERED A TWO;CHR$(34)
7630 PRINT:60 STOP
7640 PRINT:70 PRINT "CHR$(34)";"YOU ENTERED A THREE;CHR$(34)
7650 PRINT:80 STOP
7660 RETURN
7670 REM
7680 RUN*TEST2*
7690 RUN*MENU*
7700 RUN*LESSON2*
7710 END
***** Listing of Program 'TEST2' *****

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$(i) = NAMES ARRAY, USED TO READ IN SEQUENTIAL NAMES, AND TO WRITE OUT
80 REM ** S(i) = SCORES ARRAY - USED TO READ AND UPDATE NAMES.
90 REM ** D(i) = ARRAY TO KEEP TRACK OF NUMBER OF CORRECT ANSWERS. IF AN ARRAY ELEMENT EQUALS 1, THE ANSWER WAS CORRECT
100 REM **
110 CLEAR 109
120 DIM N$(10000)
130 DIM S(10000)
140 DIM D(10000)
150 PRINT* : PRINT"FINAL TEST (lesson 2)"
160 PRINT* : PRINT"This test consists of 10 questions. You must get 70 percent"
170 PRINT"of them correct to pass. (that's 7 right out of the 10 ques"
180 PRINT"tions). Use only capital letters in your answers, don't"
190 PRINT": include extra spaces or letters. GOOD LUCK"
200 PRINT* : GOSUB 3281
210 PRINT* : GOSUB 2283
220 PRINT* : PRINT"Which of the following is a legal filename?"
230 PRINT* : SODIIXE.COM (in CPM or Cromeco)
240 PRINT* : "TRIUMPH" (in TRS-80)
250 PRINT* : THEWAYOF.S21 (in CPM or Cromeco)
260 PRINT* : SCORE"DAT (in TRS-99)"
270 PRINT* : GOSUB 3281
280 INPUT"ENTER the letter opposite the correct answer"IT*$
290 PRINT* : GOSUB 2283
300 IF T$ = "C" THEN GOTO 470
310 PRINT"WRONG - the correct answer is C"
320 PRINT*: Answers A and B filenames or extensions start with"
330 PRINT*: numbers instead of alphabet characters. Answer D"
340 PRINT*: has a non alpha-numeric character in it."
350 PRINT*: See part 1, filenames, in lesson 2."
360 GOTO 470

119
***** Listing of Program 'TEST2' *****

470 PRINT"CORRECT"
480 G12 = 1
490 PRINT
500 INPUT"press ENTER":T$
510 GOSUB 7280
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$(34)'LOVE'CHR$(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 G12 = 1
700 PRINT
710 INPUT"press ENTER":T$
720 GOSUB 7280
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 G13 = 1
880 PRINT
890 INPUT"press ENTER":T$
900 GOSUB 7280
910 PRINT"Which of the following examples is INVALID?"
920 PRINT
Listing of Program 'TEST2'

930 PRINT"A" INPUT T$
940 PRINT"B" INPUT"CHR$(34)"ENTER YOUR NAME"CHR$(34)";N"
950 PRINT"C" INPUT N" INPUT
960 PRINT"D" INPUT"CHR$(34)"ENTER YOUR AGE"CHR$(34)";N"
970 PRINT
980 INPUT*ENTER the letter opposite the correct answer";T$
990 PRINT
1000 IF T$ = "B" THEN GOTO 1050
1010 PRINT"WRONG - statement B is the bad one because it tries"
1020 PRINT"to load a numeric variable with string data."
1030 PRINT"See Part I."
1040 GOTO 1070
1050 PRINT"CORRECT"
1060 S1: I = 1
1070 PRINT
1080 INPUT*press ENTER";T$
1090 GOSUB 3280
1100 PRINT*What command will let you put your program into permanent"
1110 PRINT*storage so that you can recall it later?"
1120 PRINT
1130 PRINT"A" SAVE"
1140 PRINT"B" RUN"
1150 PRINT"C" LOAD"
1160 PRINT"D" STORE"
1170 PRINT
1180 INPUT*ENTER the letter opposite the correct answer";T$
1190 PRINT
1200 IF T$ = "A" THEN GOTO 1260
1210 PRINT"WRONG - the correct answer is A"
1220 PRINT"RUN causes program execution, LOAD loads the"
1230 PRINT"program from disk and STORE is not a BASIC word."
1240 PRINT"See Part I."
1250 GOTO 1280
1260 PRINT*CORRECT"
1270 Q5: I = 1
1280 PRINT
1290 INPUT*press ENTER";T$
1300 GOSUB 3280
1310 PRINT*Which of the following statements is legal?"
1320 PRINT
1330 PRINT"A" READ A @ B @ C"
1340 PRINT"B" DATA A B C"
1350 PRINT"C" READ C, D, E,F"
1360 PRINT"D" DATA "CHR$(34)"QUIT"CHR$(34)"."CHR$(34)"FIKE"CHR$(34)"
1370 PRINT
1380 INPUT*ENTER the letter opposite the correct answer";T$

121
**** Listing of Program 'TEST2' ****

1370 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 1660
1660 PRINT"CORRECT"
1670 Q(7) = 1
1680 PRINT
1690 INPUT"press ENTER":T$
1700 GOSUB 3280
1710 PRINT"The IF statement is a CONDITIONAL BRANCHING statement."
1720 PRINT
1730 PRINT"A TRUE"
1740 PRINT"B FALSE"
1750 PRINT
1760 INPUT"ENTER the letter opposite the correct answer":T$
1770 PRINT
1780 IF T$ = "A" GOTO 1820
1790 PRINT"WRONG - the correct answer is A"
1800 PRINT"See Part 2, IF and GOTO."
1810 GOTO 1840
1820 PRINT"CORRECT"
1830 Q(8) = 1
1840 PRINT
**** Listing of Program 'TEST2' ****

1850 INPUT"press ENTER";T$
1860 GOSUB 2280
1870 PRINT"The following program will NOT have an output - TRUE or FALSE"*
1880 PRINT
1890 PRINT"10 N = 5"
1900 PRINT"20 IF N = 10 GOTO 40, 50, 60"
1910 PRINT"30 STOP"
1920 PRINT"40 PRINT N"*
1930 PRINT"50 STOP"
1940 PRINT"60 PRINT N"*
1950 PRINT"70 STOP"
1960 PRINT"80 PRINT N"
1970 PRINT"90 STOP"
1980 PRINT
1990 PRINT"A TRUE"
2000 PRINT"B FALSE"
2010 INPUT"ENTER the letter opposite the correct answer";T$
2020 PRINT
2030 IF T$ = "A" GOTO 2090
2040 PRINT"WRONG - the correct answer is A"
2050 PRINT"Line 10 sets N to 5. Line 20 only has 3 places"
2060 PRINT"to go to, so it defaults to the line under it."
2070 PRINT"That line is a STOP statement. See part 2, GOTO."
2080 GOTO 2110
2090 PRINT"C CORRECT"
2100 G1P = 'B'
2110 PRINT
2120 INPUT"press ENTER";T$
2130 GOSUB 2280
2140 PRINT"The following program will have an output - TRUE or FALSE"*
2150 PRINT
2160 PRINT"10 N = 3"
2170 PRINT"20 IF N = 3 THEN GOTO 40"
2180 PRINT"30 PRINT N"
2190 PRINT"40 N = 4"
2200 PRINT"50 END"
2210 PRINT
2220 PRINT
2230 PRINT"A TRUE"
2240 PRINT"B FALSE"
2250 PRINT
2260 INPUT"ENTER the letter opposite the correct answer";T$
2270 PRINT
2280 IF T$ = "B" GOTO 2340
2290 PRINT"WRONG - the correct answer is B"
2300 PRINT"Line 10 sets N to 3. Line 20 causes the program to"
**** Listing of Program 'TEST2' ****

2210 PRINT* go to line 40, then 50.*
2220 PRINT* See part 2, IF.*
2230 GOTO 2250
2240 PRINT'CORRECT'
2250 Y1(X) = 1
2260 PRINT
2270 INPUT'dress ENTER';T$
2280 GOSUB 3290
2290 FOR I = 1 TO 10
2300 Y = +Q(I)
2310 NEXT X
2320 PRINT'tyou have finished the test, out of 10 possible correct answers'
2330 PRINT'tyou 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 2300
2450 PRINT
2460 PRINT'Do you want your score recorded on a permanent file?'
2470 PRINT
2480 PRINT'YES'
2490 PRINT'NO'
2500 PRINT
2510 INPUT'Which';T$
2520 IF T$ = '9' THEN GOTO 2220
2530 GOSUB 2220
2540 PRINT'To record your score, we must open a file and put your name in it. Therefore, surprisingly, we need your name. If your name is not unique among the students likely to take this test, please contact your test monitor for an identifying word that will make you unique. Then enter that word below.'
2550 PRINT
2560 PRINT
2570 IF T$ = '9' THEN GOTO 2920
2580 GOSUB 2920
2590 PRINT'To record your score, we must open a file and put your name in it. Therefore, surprisingly, we need your name. If your name is not unique among the students likely to take this test, please contact your test monitor for an identifying word that will make you unique. Then enter that word below.'
2600 PRINT
2610 PRINT
2620 IF T$ = '9' THEN GOTO 2920
2630 GOSUB 2920
2640 PRINT'To record your score, we must open a file and put your name in it. Therefore, surprisingly, we need your name. If your name is not unique among the students likely to take this test, please contact your test monitor for an identifying word that will make you unique. Then enter that word below.'
2650 PRINT
2660 PRINT
2670 IF T$ = '9' THEN GOTO 2920
2680 GOSUB 2920
2690 PRINT
2700 PRINT
2710 PRINT
2720 INPUT'ENTER your word or name now';T$
2730 OPEN*"i.1","SCOREZ" 
2740 I = 0
2750 IF EOF(1) THEN GOTO 2820
***** Listing of Program 'TEST2' *****

2770 I = I+1
2780 INPUT#1,N#(I):
2790 INPUT#1,S(I):
2800 IF N#(I) = T$ THEN GOTO 2970
2810 GOTO 2750
2820 CLOSE
2830 I = I+1
2840 N#(I) = T$
2850 S(I) = Y
2860 OPEN D:", I,"SCORE2"
2870 FOR W = 1 TO I
2880 PRINT$1,N#(W)
2890 PRINT$1,S(I)
2900 NEXT W
2910 PRINT
2920 PRINT"You are now qualified to go to LESSON 3, however, you will"
2930 PRINT"first get a homework assignment! The homework is in a"
2940 PRINT"program module. If you don't want the assignment, you may"
2950 PRINT"go to the MENU instead."
2960 GOTO 2720
2970 S(I) = Y
2980 IF EOF(I) THEN CLOSE;GOTO 2000
2990 I = I+1
3000 INPUT#1, N#(I), S(I)
3010 GOTO 2790
3020 OPEN D:", I,"SCORE2"
3030 FOR W = 1 TO I
3040 PRINT$1,N#(W)
3050 PRINT$1,S(I)
3060 NEXT W
3070 PRINT
3080 PRINT"You may now take LESSON 3. You will be returned to the MENU"
3090 PRINT"from where you may go to LESSON 3 or quit."
3100 GOTO 2020
3110 IF Y=10 THEN RETURN
3120 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
3130 PRINT
3140 IF Q(1) = 0 THEN PRINT" part 1. Filehames"
3150 IF Q(2) = 0 OR Q(5) = 0 THEN PRINT" part 1. SAVE, LOAD, RUN"
3160 IF Q(3) = 0 THEN PRINT" part 1. REMarks"
3170 IF Q(4) = 0 THEN PRINT" part 1. INPUT Statements"
3180 IF Q(6) = 0 OR Q(7) = 0 THEN PRINT" part 2. READ and DATA Statements"
3190 IF Q(8) = 0 OR Q(9) = 0 THEN PRINT" part 2. IF Statements"
3200 IF Q(10) = 0 THEN PRINT" part 2. GOTO Statements"
3210 PRINT
3220 INPUT"press ENTER":I$
****** Listing of Program 'TEST2' ******

1200 GOSUB 1320
1210 RETURN
1220 REM **
1230 REM ** this subroutine clears the screen*
1240 REM **
1250 FOR X = 1 TO 24
1260 PRINT
1270 NEXT X
1280 RETURN
1290 PRINT
1300 INPUT 'press ENTER to return to the MENU':T$ 1310 RUN **MENU**
1320 PRINT
1330 INPUT 'Do you want the assignment (Y/N)':T$ 1340 IF T$ = "N" THEN GOTO 1330
1350 RUN **HW2**
10 REM ** THIS PROGRAM STARTED ON 1 MAY 1983
20 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
30 REM ** TITLE: LESSON 3
40 REM **
50 REM **
60 GOSUB 5590
70 PRINT "LESSON: BASIC" 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 LOPPS 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 5590
270 PRINT "A  I'm taking this part in its entirety."
280 PRINT "B  I wish to review selected areas."
290 PRINT "C  I want to go to the second part."
300 PRINT "D  I want to return to the Menu."
310 PRINT
320 INPUT "press either capital A, B, C, or D and then press ENTER": T2
330 IF T2 = "A" GOTO 6370
340 IF T2 = "B" GOTO 6380
350 IF T2 = "C" GOTO 460
360 IF T2 = "D" GOTO 270
370 GOSUB 570
380 GOSUB 1090
390 GOSUB 1450
400 GOSUB 2240
410 GOSUB 3400
420 PRINT
430 PRINT
440 PRINT "Going to second half of lesson 3 - Wait one second"
450 GOTO 6380
460 GOSUB 5590
**Listing of Program 'LESSON3'**

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, 1240, 1340
560 GOTO 460
570 GOSUB 5590
580 PRINT "Introduction & KILL Statement"
590 PRINT
600 PRINT "By now you should have saved a few small programs to disk, and"
610 PRINT "you should have practiced all the commands we have discussed to"
620 PRINT "date. If you have wondered how to get rid of a file that"
630 PRINT "you were done with, or have mistakenly saved and didn’t need"
640 PRINT "one of your programs, the next frame will be of service to"
650 PRINT "you. It describes the KILL statement. We have purposely"
660 PRINT "put the KILL statement in the third lesson because it is a"
670 PRINT "dangerous command that can eliminate valuable and irreplacable"
680 PRINT "data or programs if used incorrectly. By now you should feel"
690 PRINT "comfortable with some of the commands and you shouldn’t make"
700 PRINT "the mistake of KILLing someone else’s files, or your own by"
710 PRINT "accident. Remember, KILL does just what it says, it KILLS"
720 PRINT "files."
730 PRINT
740 INPUT "Press ENTER to learn about the KILL command": I$
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": I$

129
***** 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 TS = "C" GOTO 1050
1010 PRINT"WRONG - the correct answer is C"
1020 GOTO 1040
1030 PRINT"CORRECT"
1040 PRINT
1050 INPUT"press ENTER":TS
1060 GOSUB 5280
1070 IF TS = "B" GOTO 570
1080 RETURN
1090 GOSUB 5590
1100 PRINT"
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 8, the middle column"
1240 PRINT"would be the square of the figure in the adjacent first column,"
1250 PRINT"and the third column would be the square of the figure in the"
1260 PRINT"adjacent middle column. It would look like this:"
1270 PRINT
1280 FOR I = 1 TO 8
1290 PRINT INT(I),INT(I*2),INT(I*2+2)
1300 NEXT I
1310 PRINT
1320 INPUT"press ENTER":TS
1330 GOSUB 5590
1340 FOR I = 1 TO 8
1350 PRINT INT(I),INT(I*2),INT(I*2+2)
1360 NEXT I

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

1370 PRINT
1380 PRINT"If you were to write separate print statements for these"
1390 PRINT"calculations, you would have to write at least 8 lines of code."
1400 PRINT"Using LOOPS, you can calculate this column of figures, and"
1410 PRINT"display it, in three lines of code!"
1420 PRINT
1430 INPUT"In the next sections we will learn this and more...press ENTER":T$ 1440 RETURN
1450 GOSUB 5590
1460 PRINT" COUNTER variables"
1470 PRINT
1480 PRINT"The first step in learning about loops is to understand COUNTER"
1490 PRINT"variables. A COUNTER is a variable that is used to keep track"
1500 PRINT"of the number of times a program executes a line. It is "
1510 PRINT"often used in conjunction with an IF THEN statement. For "
1520 PRINT"example:"  
1530 GOSUB 6050
1540 PRINT"The value of I after running this program is 10."
1550 PRINT
1560 INPUT"press ENTER":T$ 1570 GOSUB 5590
1580 PRINT" COUNTERs (cont)"
1590 PRINT
1600 GOSUB 6050
1610 PRINT"The COUNTER variable is I. It is INCREMENTED every time line"
1620 PRINT"50 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":T$ 1670 GOSUB 5590
1680 GOSUB 6050
1690 PRINT"Do you see what would happen if we didn't include the IF"  
1700 PRINT"statement? The program would look like this:"  
1710 PRINT
1720 PRINT"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, HUNG, and *** DUMMY. you did it again! ***"  
1790 PRINT
1800 INPUT"press ENTER":T$ 1810 GOSUB 5590
1820 PRINT"
1830 PRINT" COUNTERs cont:"
1820 PRINT
1830 PRINT "If your program ever gets hung up, you may have to type "
1840 PRINT "CONTROL-C (if CPM or Cromemco), or BREAK (if TRS-80)."
1850 PRINT "In some extreme cases, you may have to reset the system."
1860 PRINT "Do the TRS-80, the RESET button is the red recessed button.
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 reset the system, let the"
1900 PRINT "operator help you fix the problem. If you are using a single"
1910 PRINT "user system, such as a TRS-80, then RESET will work as an exit"
1920 PRINT "from the locked up program (but try BREAK first). If you use"
1930 PRINT "either RESET or CONTROL-C, the system will take you to the"
1940 PRINT "COMMAND mode, and you will have to re-initialize BASIC."
1950 PRINT
1960 INPUT "press ENTER to continue..."
1970 INPUT "press ENTER to continue..."
1980 INPUT "What is the COUNTER variable in this program?"
1990 PRINT
2000 PRINT "I = 0"
2010 PRINT "I = " : I = I + 1"*
2020 PRINT "I = 1 THEN I = 2"
2030 PRINT "STOP"
2040 PRINT "STOP"
2050 PRINT "GOTO 20"
2060 PRINT "END"
2070 PRINT "I is the COUNTER"
2080 PRINT "I is the COUNTER"
2090 PRINT "BEH are COUNTERS, but I will control the program"
2100 PRINT
2110 INPUT "Press the letter opposite the correct answer and press ENTER to continue..."
2120 PRINT
2130 IF I = 9 THEN GOTO 519
2140 PRINT "INCINC - the correct answer is B"
2150 PRINT "INCINC"
2160 PRINT "INCINC" - that shows good understanding of the principle"
2170 PRINT
2180 IF I = 9 THEN GOTO 519
2190 RETURN
2200 "press ENTER to continue..."
2210 PRINT "FOR - NEXT Statements"
2220 PRINT
2230 PRINT "In the previous discussion of counters, we showed you that the"
***** Listing of Program 'LESSON' *****

1290 PRINT"tests a counter with an IF statement to see if it had reached"
1291 PRINT"its desired level. If it had, then we went to another part of the"
1292 PRINT"PROGRAM, or STOPped. There is a set of statements in BASIC"
1293 PRINT"that lets us shorten the statements needed to duplicate this:
1294 PRINT" Đầu of program. The set of statements is called FOR - NEXT."
1295 PRINT
1296 PRINT
1297 INPUT press ENTER='1'
1298 GOSUB 9290
1299 GOSUB 9290
1300 PRINT FOR - NEXT (cont)"
1301 GOSUB 9113
1302 PRINT"The two programs above will print exactly the same output.
1303 PRINT"The COUNTER loop requires more statements and is not as"
1304 PRINT"efficient as the FOR NEXT loop."
1305 PRINT
1306 INPUT press ENTER='1'
1307 GOSUB 9290
1308 GOSUB 9290
1309 PRINT"Line 10 of the FOR NEXT loop initializes i, the same way that"
1310 PRINT"lines 10 and 20 of the COUNTER loop do. The FOR NEXT loop sees"
1311 PRINT"line 10 as 'I'm going to start a loop. i is the counter. I will'
1312 PRINT"begin with i = 1 and when it is GREATER than 3 I will GOTO the'
1313 PRINT"statement that is after the NEXT i statement."
1314 PRINT
1315 INPUT press ENTER='1'
1316 GOSUB 9290
1317 GOSUB 9290
1318 GOSUB 9290
1319 PRINT"are the same and perform the same function."
1320 PRINT"Note that the PRINT statement was indented a few spaces in the"
1321 PRINT"FOR NEXT loop. This lets you see the loop structure better."
1322 PRINT"You should do that in your own programming, too."
1323 PRINT
1324 INPUT press ENTER='1'
1325 GOSUB 9290
1326 GOSUB 9290
1327 PRINT"The IF statement in line 40 of the COUNTER loop determines if i"
1328 PRINT"has reached 3. This was done automatically by the FOR NEXT"
1329 PRINT"loop because the last value in line 20 was specified as 3."
1330 PRINT"Line 50 of the FOR NEXT loop is the same as the UNCONDITIONAL"
1331 PRINT"GOTO in line 50 of the COUNTER loop. The NEXT i statement"
1332 PRINT"increments i and sends control to the FOR statement."
1333 INPUT in this case control goes to line 1..............press ENTER='1'
1334 GOSUB 9290
1335 PRINT"IF i = value #1, to value #2:"
1336 PRINT"END OF PROGRAM"
**** Listing of Program 'LESSON' ****

07/10/82 - 11:02:39

1780 PRINT"DO NEXT""
1790 PRINT"
1800 PRINT"In this... the FOR X = :value #:1; to :value #:2"
1810 PRINT"causes the variable X to be initially set at value #:1."
1820 PRINT"and the program executes the next lines until it reaches the".
1830 PRINT"END NEXT". It is an example, any VARIABLE works. The".
1840 PRINT"NEXT statement causes an UNCONDITIONAL GOTO to the FOR state-
1850 PRINT"ment. The variable is incremented and the FOR word tests to".
1860 PRINT"see if it EXCEEDS value #:2. IF it does, then control passes".
1870 PRINT"to the statement that follows the NEXT statement."
1880 PRINT"In this - there are no statements passed the NEXT word, the program".
1890 PRINT"END NEXT".".
1900 PRINT"
1910 FOR I = 1 TO 5
1920 PRINT X+I
1930 NEXT I
1940 PRINT"
1950 PRINT"Which of the following programs made the above list?"
1960 PRINT"
1970 PRINT"END NEXT 1. FOR X=1 TO 5 0 10 FOR I = 1 TO 5"
1980 PRINT* 20 PRINT X+I 30 PRINT X+2".
1990 PRINT* 40 NEXT I 50 NEXT I".
2000 PRINT"
2010 PRINT B 10 X=I+I 0 10 I=I+1"
2020 PRINT* 20 PRINT X+I 30 PRINT X+2".
2030 PRINT* 40 IF x=I THEN GOTO 1) 50 IF I=4 THEN STOP"
2040 PRINT* 60 GOTO 10 40 GOTO 10"
2050 PRINT"
2060 INPUT'press the letter opposite the correct choice and press ENTER*"*
2070 PRINT"
2080 IF 'A' = 'B' THEN GOTO 10"
2090 PRINT"ERROR - the correct answer is A"
2100 GOTO 10"
2110 PRINT"CORRECT"
212 PRINT
LESSON 8.1

INPUT press ENTER
210 GOSUB 200
215 PRINT "The NEXT statement is the same as an UNCONDITIONAL BRANCHING"
220 PRINT "Statement. It branches to the next line under the FOR statement."
225 PRINT "It sets the variable before the FOR statement tests it."
230 PRINT
235 PRINT IS the paragraph above TRUE or FALSE?
240 INPUT " TRUE, FALSE"
245 PRINT
250 IF " $ = " THEN GOTO 220
255 PRINT "Wrong - the NEXT statement is UNCONDITIONAL, but itbranches"
260 PRINT to the SAME line as the FOR statement. It also incre-
265 PRINT sets the variable before the FOR statement tests it."
270 GOTO 220:
275 PRINT "CORRECT"
280 PRINT
285 INPUT press ENTER
290 GOSUB 200
295 IF $ = " $ THEN GOTO 220:
300 RETURN
305 GOSUB 200
310 PRINT Advanced FOR NEXT*
315 PRINT
320 PRINT FOR $ = 1 TO 10 STEP 2
325 PRINT PRINT NEXT*
330 PRINT FOR $ = 1 TO 10 STEP 2
335 PRINT PRINT NEXT*
340 PRINT =
345 NEXT
350 PRINT
355 INPUT "Interesting. isn't it? .... press ENTER for an explanation:"*
360 GOSUB 200:
365 PRINT Advanced FOR NEXT*
370 PRINT
375 PRINT FOR $ = 1 TO 10 STEP 2:
380 PRINT PRINT NEXT*
385 PRINT FOR $ = 1 TO 10 STEP 2:
390 PRINT PRINT NEXT*
395 PRINT
400 PRINT "The STEP word in the FOR statement caused the value of $ to

#### Listing of Program 'LESSON 2' ####

325 PRINT 'The output of 1, 2, 3, 4, 5 was'
326 PRINT 'Correct because the NEXT statement is the statement that'
327 PRINT 'Actually increments the value, so the first time through,'
328 PRINT 'Was equal to 1, the next time through it was equal to 2, etc.'
329 PRINT
330 INPUT 'Press ENTER for explanation'
331 GOSUB 5290
332 PRINT 10 FOR x = 10 TO 1 STEP -2
333 PRINT 20 PRINT x
334 PRINT
335 PRINT 'In this case, we stepped down instead of up. Notice that the'
336 PRINT 'First value of the FOR statement is the largest value. It'
337 PRINT 'Would not make sense to specify a negative STEP and give the'
338 PRINT 'Range of a positive STEP. For example: '
339 PRINT
340 PRINT 10 FOR x = 1 TO 10 STEP -1
341 PRINT
342 PRINT 'This does not make sense and would not work.'
343 PRINT
344 GOSUB 5290
345 INPUT 'Press ENTER'
346 PRINT 'You may include a FOR NEXT statement within another FOR NEXT'
347 PRINT 'Statement. If you do, it is called NESTED looping.'
348 PRINT 'For example:'
349 GOSUB 6290
350 PRINT 'Before we explain the output, do you see how we indented the'
351 PRINT 'Statements within the first loop, and then further indented'
352 PRINT 'The statements within the second loop? It makes them easier'
353 PRINT 'To understand (and explain).'
*** Listing of Program 'LESSON3' ***

4100  GOSUB 5590  
4105  PRINT"Advanced FOR NEXT (NESTED LOOPS):"
4110  PRINT
4115  FOR  X = 1 TO 2
4120  FOR  Y = 1 TO 2
4125  PRINT  X,Y
4130  NEXT  Y
4135  NEXT  X
4140  PRINT
4145  GOSUB 6290
4150  PRINT"Try to figure how the program produced the columns above it"
4155  INPUT"and then press ENTER":T$
4160  GOSUB 5590
4165  PRINT
4170  FOR  X = 1 TO 2
4175  FOR  Y = 1 TO 2
4180  PRINT  X,Y
4185  NEXT  Y
4190  NEXT  X
4195  PRINT
4200  GOSUB 5590
4205  INPUT"Try to figure how the program produced the columns above it"
4210  INPUT"and then press ENTER":T$
4215  GOSUB 6290
4220  PRINT
4225  FOR  X = 1 TO 2
4230  FOR  Y = 1 TO 2
4235  PRINT  X,Y
4240  NEXT  Y
4245  NEXT  X
4250  PRINT"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."
4255  PRINT
4260  INPUT"Can you figure it out now? Think about it and then press ENTER":T$
4265  GOSUB 5590
4270  GOSUB 6290
4275  PRINT"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."
4280  PRINT
4285  INPUT"Can you figure it out now? Think about it and then 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":T$  
4700 GOSUB 5590
4710 PRINT"Is the following program valid?"  
4720 PRINT
4730 PRINT"10 FOR I = 1 to 200"  
4740 PRINT"20 FOR J = 1 to 2"  
4750 PRINT"30 PRINT I=2"  
4760 PRINT"40 NEXT 2"  
4770 PRINT"50 NEXT 2"  
4780 PRINT
4790 PRINT"A YES it is a good program"  
4800 PRINT"B NO it is not a valid program"  
4810 PRINT
4820 INPUT"Enter the letter opposite the correct answer":T$  
4830 PRINT
4840 IF T$ = "A" GOTO 4860
4850 PRINT"Wrong - the correct answer is A. There is nothing wrong."  
4860 PRINT"with the program."  
4870 GOTO 4900
4880 PRINT"CORRECT!"  
4890 PRINT
4900 INPUT"Press ENTER":T$  
4910 GOSUB 5590
4920 PRINT"Give the first statement of a FOR NEXT loop, that will:"  
4930 PRINT"cause the loop to increment the variable I from one to 20 in"  
4940 PRINT"steps of 2. Use 10 as your line number and leave one space"  
4950 PRINT"between all words, numbers, and variables."  
4960 PRINT
4970 INPUT"Enter your answer":T$  
4980 PRINT
4990 IF T$ = "10 FOR I = 1 TO 20 STEP 2" GOTO 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
5120 INPUT"press ENTER":TS
5060 GOSUB 5590
5070 PRINT"is the following program valid?"
5080 PRINT
5100 PRINT"FOR I = 1 TO 2"
5120 PRINT"FOR I = 1 TO 5"
5140 PRINT"CHR$(34)" "This is the last question" CHR$(34)
5160 PRINT"NEXT I"
5180 PRINT"NEXT I"
5200 PRINT"PRINT A Yes, it is valid"
5220 PRINT"PRINT B No, it is not valid"
5240 PRINT"PRINT C This is the last question"
5260 PRINT"EXIT"
5280 PRINT"NEXT I"
5300 PRINT"PRINT "CORRECT! - and the output is shown above"
5320 PRINT
5340 INPUT"press ENTER":TS
5360 GOSUB 5590
5380 PRINT"Remember in the beginning of this section when we mentioned"
5400 PRINT"the program that produced three columns of figures, the first"
5420 PRINT"column was integer i through 8, the second column was the "
5440 PRINT"square of the first, and the third the square of the second"
5460 PRINT"Think you can figure out how we did it?"
5480 PRINT
5500 INPUT"press ENTER for the answer":TS
5520 GOSUB 5590
5540 FOR I = 1 TO 3
5560 PRINT I, I*I, (I%1)*(I%1)
5580 NEXT I
5600 PRINT
5620 PRINT"FOR I = 1 TO 3"
5640 PRINT I, I*I, (I%1)*(I%1)
5660 PRINT"EXIT"
**** Listing of Program 'LESSON3' ****  

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 5430
5550 RETURN
5560 REM **
5570 REM ** This subroutine clears the screen on any terminal
5580 REM **
5590 FOR X = 1 TO 24
5600 PRINT
5610 NEXT X
5620 RETURN
5630 REM **
5640 REM ** THIS ROUTINE IS THE MENU
5650 REM **
5660 PRINT "LESSON 3"
5670 PRINT
5680 PRINT "This is the first part of a two part lesson"
5690 PRINT "It is divided into the following sections."
5700 PRINT
5710 PRINT1: Introduction & kill
5720 PRINT2: Loops (Intro)
5730 PRINT3: 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 5830
5890 REM **
5900 REM ** THIS IS EXAMPLE PROGRAM TO SHOW HOW A LOOP WORKS
5910 REM **
5920 PRINT
5930 PRINT "Loops (Intro)"
5940 PRINT "This program calculates a table of squares of numbers"
5950 PRINT "x = y"
***** Listing of Program "LESSON3" *****

570 PRINT"I = X + 1"
5710 PRINT"40 PRINT I, I*1, (I*1)* (I*1)"
5720 PRINT"50 IF I = 9 THEN GOTO 70"
5730 PRINT"60 GOTO 20"
5740 PRINT"70 END"
6000 PRINT
6010 RETURN
6020 REM
6120 REM This example program for COUNTER section
6130 REM
6139 PRINT
6140 PRINT"10 Z = 0"
6150 PRINT"20 Z = Z + 1"
6160 PRINT"30 IF Z = 10 THEN STOP"
6170 PRINT"40 GOTO 20"
6180 PRINT
6190 RETURN
6200 REM
6210 REM This example is for the FOR NEXT section
6220 REM
6229 PRINT
6230 PRINT"COUNTER loop FOR NEXT loop"
6240 PRINT
6250 PRINT"10 FOR I = 1 TO 3"
6260 PRINT"20 PRINT I"
6270 PRINT"30 PRINT X"
6280 PRINT"40 IF X = 3 GOTO 50"
6290 PRINT"50 GOTO 20"
6300 PRINT"60 END"
6310 PRINT
6320 RETURN
6330 REM
6340 REM This example is for the Advanced FOR NEXT section
6350 REM
6359 PRINT
6360 PRINT"10 FOR X = 1 TO 2"
6370 PRINT"20 FOR Y = 1 TO 2"
6380 PRINT"30 PRINT X, Y"
6390 PRINT"40 NEXT Y"
6400 PRINT"50 NEXT X"
6410 PRINT
6420 RETURN
6430 RUN "MENU"
6440 RUN "LESSON9A"
6450 END
250  GO SUB 4090
260  GO SUB 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$ = "B" GOTO 420
360  IF T$ < "A" GOTO 270
370  GO SUB 540
380  GO SUB 1260
390  GO SUB 2430
400  GO SUB 3950
410  GO SUB 4080
420  GOTO 4760
430  GO SUB 4080
440  GO SUB 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 .1360 .2430 .3950
530  GOTO 430
540  GO SUB 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  GO SUB 4090
670  PRINT" Arrays (Intro)"
680  PRINT
690  PRINT
700  PRINT" One use for ARRAYS would be to make a training program that "
710  PRINT" listed the people on training, their time in the organization."
720  PRINT
**** Listing of Program 'LES10ZIA' ****  

650 PRINT "or rank, and their training status. We could group the major visual 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 700 INPUT "press ENTER"; T$
710 GOSUB 4080
720 GOSUB 4410
730 PRINT "We could assign subscripted ARAYS to the three main categories. An ARRAY has the following format:" 740 PRINT 750 PRINT 760 PRINT 770 PRINT 780 PRINT 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 900 PRINT 910 INPUT "press ENTER"; T$
920 GOSUB 4080
930 PRINT 940 PRINT 950 PRINT 960 PRINT 970 PRINT 980 PRINT 990 PRINT
1000 PRINT 1010 PRINT 1020 PRINT 1030 PRINT 1040 PRINT 1050 PRINT 1060 PRINT 1070 INPUT "This program, when RUN, would print our data.....press ENTER"; T$
1080 GOSUB 4080
1090 GOSUB 4410

N$O) = "CHR$(34)"John Doe"CHR$(34)"
N$1) = "CHR$(34)"Jake Robinski"CHR$(34)"
N$2) = "CHR$(34)"Mark Muffin"CHR$(34)"
R$(O) = "CHR$(34)"Foreman"CHR$(34)"
R$(1) = "CHR$(34)"Peon"CHR$(34)"
R$(2) = "CHR$(34)"Specialist"CHR$(34)"
T(0) = "CHR$(34)"Nase Rank Training Status"CHR$(34)"
T(1) = "CHR$(34)"Nase Rank Training Status"CHR$(34)"
T(2) = "CHR$(34)"Nase Rank Training Status"CHR$(34)"
T(3) = "CHR$(34)"Nase Rank Training Status"CHR$(34)"
1100 PRINT"This is what that program would output. Notice that we have"
1110 PRINT"used only three variables, but we made them ARRAYS so that we"
1120 PRINT"could hold nine bits of data."
1130 PRINT"As we continue through our lessons, we will discover some very"
1140 PRINT"powerful uses for ARRAYS."
1150 PRINT
1160 INPUT"press ENTER";T$  
1170 GOSUB 4080
1180 PRINT"Is the following ARRAY and its subscript valid?"
1190 PRINT
1200 PRINT"A(0)"
1210 PRINT"B TRUE"
1220 PRINT"C 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 4080
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"Microscript, will handle at least 8 dimensions. An example of a"
1490 PRINT"TWO dimension ARRAY would be M(2,2). Notice that the extra"
1490 PRINT"dimension was designated by just adding another subscript in-
1500 PRINT"side the parenthesis. A THREE DIMENSIONED ARRAY looks like"
1510 PRINT"this: R(2,1,9) or T(5,4,3) (or any combination of numbers)."
1520 INPUT"press ENTER";T$  
1530 GOSUB 4080
1540 GOSUB 4520

143
1550 PRINT "The above program combines several of the techniques that we"
1560 PRINT "have been learning. Before we tell you the answer, try to "
1570 PRINT "figure out what the output of the program would be. We warn"
1580 PRINT "you, it is a little tricky, but see if you can figure it out."
1590 PRINT
1600 INPUT "press ENTER*":T$
1610 GOSUB 4080
1620 GOSUB 4520
1630 PRINT 5 10 15 20
1640 PRINT
1650 PRINT "This is the output. Lines 10 and 20 are nested FOR NEXT loops."
1660 PRINT "They set up the READ statement in line 30 so that it will READ"
1670 PRINT "the values that are in the DATA statement and assign the"
1680 PRINT "current X, Y subscript to it. Values are read in one at a time."
1690 INPUT "press ENTER*":T$
1700 GOSUB 4080
1710 GOSUB 4520
1720 PRINT "On the first pass, X = 1 and Y = 1. A(1,1) therefore, equals 5."
1730 PRINT "The semi-colon on the end of the print statement causes the "
1740 PRINT "numbers to be printed side by side instead of on separate lines"
1750 PRINT "and they all have one space between them. (caused by the ":")"
1760 PRINT "On the second iteration of Y, Y will equal 2 and X = 1. A(1,2) "
1770 PRINT "will equal 10."
1780 PRINT
1790 INPUT "press ENTER*":T$
1800 GOSUB 4080
1810 GOSUB 4520
1820 PRINT "When the second iteration of Y is done, control will pass to "
1830 PRINT "line 10 and X will begin its second iteration. The Y loop"
1840 PRINT "will start all over again and when line 30 is executed (3rd"
1850 PRINT "time), A(1,2) will equal 15. Finally, Y will execute for the"
1860 PRINT "fourth time (second time while X = 2), and A(X,Y) will have all"
1870 PRINT "numbers in the DATA statement, and the numbers will have "
1880 PRINT "printed out."
1890 PRINT
1900 INPUT "press ENTER*":T$
1910 GOSUB 4080
1920 GOSUB 4520
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, Y must equal 1."
1970 PRINT "And X must equal 2. Do you see?"
1980 PRINT
1990 INPUT "press ENTER*":T$
2000 GOSUB 4080
**** Listing of 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 T$ = "C" GOTO 2120
    2110 GOTO 2130
    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 T$ = "D" GOTO 2280
    2260 PRINT "WRONG - the correct answer is D"
    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 T$ = "S" GOTO 1360
    2420 RETURN
    2430 GOSUB 4:99
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 (i.e. N(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 I = 1 TO 12"
2620 PRINT "30 READ B(I)"
2630 PRINT "40 NEXT I"
2640 PRINT "50 DATA 5.10.15.20.25.30.35.40.45.50.55.60"
2650 PRINT "60 FOR I = 1 TO 12"
2660 PRINT "70 PRINT B(I)"
2670 PRINT "80 NEXT I"
2680 PRINT "RUN"
2690 PRINT
2700 PRINT "The DIM statement told the computer that 12 pockets were needed"
2710 PRINT "and the READ, DATA, and PRINT statements filled the ARRAY and"
2720 PRINT "printed it out."
2730 PRINT
2740 INPUT "Press ENTER"; T$
2750 GOSUB 4080
2760 PRINT "10 15 20 25 30 35 40 45 50 55 60"
2770 PRINT
2780 PRINT "The DIM statement told the computer that 12 pockets were needed"
2790 PRINT "and the READ, DATA, and PRINT statements filled the ARRAY and"
2800 PRINT "printed it out."
2810 PRINT
2820 INPUT "Press ENTER"; T$
2830 GOSUB 4080
2840 PRINT * Dimension and Clear*
2850 PRINT
2860 PRINT "Another problem the computer has with memory allocation is"
2870 PRINT "reserving enough room for STRING space. For every letter in"
2880 PRINT "a STRING, the BASIC processor must use a little over one mem."
2890 PRINT "or space. It must be able to find the letter once it stores it"
2900 PRINT "so it uses one location to store the letter, and another to"
2910 PRINT "remind it where it put the STRING in the first place."
2920 PRINT "Unlike number variables, STRINGS can use up to 255 characters"
265: PRINT"to use more than 50 characters worth of STRING space, you must"
266 PRINT"CLEAR more room for it. The clear statement actually WIPES OUT"
267 PRINT"data space and reserves memory locations, so it must ALWAYS be"
268 PRINT"the first statement if you are going to need it."
269 PRINT
270 INPUT"press ENTER";T$
271 GOSUB 4090
272 PRINT* Dimension and Clear"
273 PRINT
274 PRINT"If you use the CLEAR statement in the middle of a program,"
275 PRINT"the accumulation of data that you have stored in variables to"
276 PRINT"the point that the CLEAR word was used, will be zeroed out."
277 PRINT
278 PRINT 10 X = 150"
279 PRINT 20 CLEAR;*
280 PRINT 30 PRINT X"
281 PRINT 40 PRINT
282 PRINT 50 PRINT 0"
283 PRINT 60 PRINT
284 PRINT
285 PRINT"Do you see"; The CLEAR word zeroed out X'.....press ENTER";T$
286 PRINT "GOSUB 4090"
287 PRINT 10 CLEAR 1000"
288 PRINT 20 DIM A$(100);* A$(100)*
289 PRINT 30 FOR X = 1 TO 100"
290 PRINT 40 READ A$(X)*
291 PRINT 50 PRINT ......etc.*
292 PRINT
293 PRINT"The above shows the first 4 lines of a program that is going"
294 PRINT"to use more than 1000 spaces of string space (that will give"
295 PRINT"an average of 10 letters per pocket of A$(X) and is going to"
296 PRINT"use 50 pockets of the numeric array S. Note that you always"
297 PRINT"set pocket X for free. It is normal, not used)!"
298 PRINT
299 PRINT
300 PRINT"Notice the position of the CLEAR and DIM statements."
301 INPUT"press ENTER";T$
302 GOSUB 4090
303 PRINT* What will be the output of the following program?"*
304 PRINT
305 PRINT 10 A$(12) = 5"
306 PRINT 20 CLEAR* *
307 PRINT 30 PRINT A$(12)*
308 PRINT
309 PRINT"A 0 because the clear statement is in the wrong place."
310 PRINT
311 PRINT* 5 - the clear statement only affects STRINGS.*

147
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 - Goodoodoodoodoodood 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 - Goodoodoodoodoodoodoodoodoodoodoodood 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(1,2)"
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(I,F). 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 "CLEAR 1000"
3770 PRINT "20 DIM A(2)"
3780 PRINT "A(2) = 5"
LISTING OF PROGRAM 'LESSON3A'  
07/10/95 - 01:13:48  

1790 PRINT"A) The DIM statement is in the wrong place."
1800 PRINT
1810 PRINT"B) The CLEAR statement is invalid."
1820 PRINT"C) Nothing."
1830 PRINT
1840 INPUT"ENTER the letter opposite the correct answer":T$
1850 PRINT
1860 IF T$ = "C" GOTO 3900
1870 PRINT"Wrong - the correct answer is C"
1880 GOTO 3910
1890 PRINT"CORRECT"
1900 PRINT
1910 INPUT"press ENTER":T$
1920 GOSUB 4290
1930 IF T$ = "B" GOTO 2470
1940 RETURN
1950 GOSUB 4080
1960 TEST
1970 PRINT*  
1980 PRINT"You have completed this lesson. ENTER a 'C' to Continue"
1990 PRINT"to the TEST or ENTER an 'R' to start over."
2000 PRINT
2010 INPUT"ENTER your choice":T$
2020 IF T$ = "C" THEN GOTO 4740
2030 PRINT
2040 REM ** This subroutine clears the screen on any terminal
2050 REM **
2060 FOR I = 1 TO 24
2070 PRINT
2080 NEXT I
2090 PRINT
2100 PRINT"LESSON 3B"
2110 PRINT"This is the second part of a two part lesson"
2120 PRINT"It is divided into the following sections."
2130 PRINT
2140 PRINT"1) Arrays (Introduction)  2) Dimension & Clear"
2150 PRINT"3) Test"
*** Listing of Program 'LESSON5A' ***

4250 RETURN
4260 REM **
4270 REM ** THIS LETS STUDENT REVIEW LESSONS AGAIN
4280 REM **
4290 GOSUB 4080
4300 PRINT"Which do you wish to do?"
4310 PRINT
4320 PRINT"A Continue on"
4330 PRINT"B Review this section again"
4340 PRINT
4350 INPUT"Press the letter opposite the correct answer and press ENTER":T$
4360 IF T$ = "A" OR T$ = "B" THEN RETURN
4370 GOTO 4290
4380 REM
4370 REM This subroutine is for the first Subscripts example
4400 REM
4410 PRINT
4420 PRINT"Name         Rank     Training Status"
4430 PRINT
4440 PRINT"John Doe     Foreman     9"
4450 PRINT"Jake Robinski Peon       3"
4460 PRINT"Mark Muffin Specialist  5"
4470 PRINT
4480 RETURN
4490 REM **
4500 REM ** THIS ROUTINE IS FOR ARRAYS EXAMPLE
4510 REM **
4520 PRINT
4530 PRINT"FOR Y = 1 TO 2"
4540 PRINT"FOR Y = 1 TO 2"
4550 PRINT"READ A(Y,Y)"
4560 PRINT"PRINT A(X,Y)"
4570 PRINT"NEXT Y"
4580 PRINT"NEXT X"
4590 PRINT"DATA 5,10,15,20"
4600 PRINT
4610 RETURN
4620 REM **
4630 REM ** THIS ROUTINE IS 2ND ARRAY EXAMPLE
4640 REM **
4650 PRINT"Y"
4660 PRINT"1.  2."
4670 PRINT"[-----I-----]"
4680 PRINT"1.  2.  3.  4."
4690 PRINT"[-----I-----] ARRAY A(X,Y)"
4700 PRINT"2.  3.  4.  5."

...
***** Listing of Program 'LESSONIA' *****

4710 PRINT; "---[-----[-----[-----[------";
4720 PRINT
4730 RETURN
4740 RUN "TEST1"
4750 RUN "MENU"
4760 RUN "LESSON1"
4770 END
**** Listing of Program 'TEST1' ****

1000 REM **
1110 REM ** LESSON: TEST1
1120 REM ** AUTHOR: CAPT DAN CREAGAN
1130 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
1140 REM **
1150 REM ** VARIABLES:
1160 REM ** N(N) = NAMES ARRAY, USED TO READ IN SEQ-
1170 REM ** UENTIAL NAMES, AND TO WRITE OUT
1180 REM ** S(N) = SCORES ARRAY - USED TO READ AND
1190 REM ** WRITE SCORES
1190 REM ** O(N) = ARRAY TO KEEP TRACK OF NUMBER OF
1190 REM ** CORRECT ANSWERS. IF AN ARRAY
1190 REM ** ELEMENT EQUALS 1, THE ANSWER WAS
1190 REM ** CORRECT
1190 REM **
1190 REM **
1190 REM ** CLEAR 2000
1200 GOSUB 4000
1210 DIM N(6500)
1220 DIM O(6500)
1230 DIM S(1000)
1240 PRINT "FINAL TEST (Lesson 11)"
1250 PRINT
1260 PRINT "This test consists of 10 questions. You must get 70 percent."
1270 PRINT "Do the best you can. that's 7 right out of the 10 ques-
1280 PRINT "tions). Use only capital letters in your answers. Don't"
1290 PRINT "include extra spaces or letters. GOOD LUCK."
1290 PRINT
1300 INPUT 'press ENTER to continue""T"
1310 GOSUB 4000
1320 PRINT "Which of the following is (are) invalid:""T"
1330 PRINT
1340 PRINT "CHR$ 24" "CHAL$ "CHR$ 44"
1350 PRINT "CHR$ 24" "CHR$ 24"
1360 PRINT "CHR$ 24" "CHR$ 24"
1370 PRINT
1380 INPUT "Enter the letter opposite the correct answer":"T"
1390 PRINT
1400 IF T = "A" GOTO 1490
1410 PRINT "Wrong - the correct answer is A."
1420 PRINT "Answer 3 needs SPACE: It should be right."
1430 PRINT "In answers 1 & 2, SPACE is not a BASIC word."
1440 PRINT "See part A - Ill."
1450 GOTO 1470
1460 PRINT "CORRECT"
**** Listing of Program 'TEST3' ****

1400 G:11 = 1
1410 PRINT
1430 INPUT"press ENTER to continue"
1450 GOSUB 4380
1600 PRINT"Does the FOR NEXT combination have to be used to set up a loop?"  
1610 PRINT
1620 PRINT"YES"  
1630 PRINT"NO"  
1640 PRINT
1650 INPUT"ENTER the letter opposite the correct answer:"
1660 PRINT
1670 IF I$ = "B" THEN GOTO 200
1690 PRINT"Wrong - the correct answer is B"  
1690 PRINT"A loop can be made with a variety of techniques."
1700 PRINT"Including counters and IF statements, FOR - NEXT, etc."
1710 PRINT"See part 1, LOOP, FOR NEXT."
1720 GOTO 150
1730 PRINT"Correct."
1740 C:1 = 1  
1750 PRINT
1760 INPUT"press ENTER to continue"
1770 GOSUB 4380
1790 PRINT"Write out the first line of a FOR NEXT loop using i as the"  
1790 PRINT"variable. Start the loop at 1 and end it at 5. Use a STEP of 2."
1800 PRINT"Use line number 50. Use all caps. Leave one space between all"  
1800 PRINT"terms."
1810 PRINT
1810 PRINT"Correct."
1820 C:1 = 1  
1830 PRINT
1840 INPUT"press ENTER to continue"
1850 GOSUB 4380
1870 PRINT"FOR X = 1 TO 5 STEP 2"  
1880 PRINT
1890 PRINT"CORRECT."
1900 PRINT"
1910 PRINT"Correct."
1920 PRINT"Enter the output from this program. Leave one space between"  
1930 PRINT"terms. Make sure to consider the semicolon on line 241"  
1940 PRINT

1950
**Listing of Program "TEST"**

1720 INPUT "ENTER your answer:"
1730 PRINT
1740 IF T5 = "A" THEN GOTO 2200
1750 PRINT "WRONG - the correct answer is: 15 E
1760 PRINT "The first time through, a '1' is printed, then the'
1770 PRINT "NEXT statement increments X by 1. The second time'
1780 PRINT "through a '3' is printed. BESIDE the 1 (because of the 'X')"
1790 PRINT "then the NEXT X statement again"
2000 PRINT "2 increments X by 2. The last time through, a 5 is printed."
2010 PRINT "See part 1. Advanced FOR - NEXT"
2020 GOTO 2200
2030 PRINT "CORRECT"
2040 D(1) = 1
2050 PRINT
2650 INPUT "press ENTER:"
2660 GOSUB 4000
2670 PRINT "How many objects (or elements) can an array have without a"
2680 PRINT "DIM statement?"
2690 PRINT
2710 PRINT "2. not counting the 0 element"
2720 PRINT "2. not counting the 0 element"
2730 PRINT "2. not counting the 0 element"
2740 PRINT "2. not counting the 0 element"
2750 PRINT
2760 INPUT "ENTER the letter opposite the correct answer:"
2770 PRINT
2770 IF "T5 = "A" THEN GOTO 2200
2790 PRINT "WRONG - the correct answer is A"
2820 PRINT "See part 2. ARRAYS"
2830 GOTO 2200
2840 PRINT "CORRECT"
2850 D(3) = 1
2860 PRINT
2950 INPUT "press ENTER:"
2960 GOSUB 4000
2970 PRINT "How many objects are there in a 2 X 3 array?"
2980 PRINT
2990 INPUT "ENTER your answer:"
3000 PRINT
3010 IF "T5 = "A" THEN GOTO 2200
3020 PRINT "WRONG - the correct answer is 6"
3030 PRINT "The easy way to count them is to multiply them."
3040 PRINT "See part 2. ARRAYS"
3050 GOTO 2200
3060 PRINT "CORRECT"
3070 D(3) = 1
100  PRINT$
110  INPUT"press ENTER"$
120  GOSUB 400$
130  PRINT"How many dimensions does the following array have?"$
140  PRINT$
150  INPUT"A"$
160  PRINT$
170  INPUT"ENTER your answer"$
180  PRINT$
190  IF A = 1 THEN GOTO 200$
200  PRINT"Wrong - the correct answer is 10"$
210  PRINT"The 10s are the first subscript of the array."
220  PRINT"See part 2. ARRAYS."
230  GOTO 190$
240  PRINT"CORRECT"
250  L = 1
260  PRINT$
270  INPUT"press ENTER"$
280  GOSUB 400$
290  PRINT"Does the following array need to be DIMensioned?"
300  PRINT$
310  PRINT M(2,1,2)$
320  PRINT$
330  INPUT"ENTER YES or NO"$
340  PRINT$
350  IF A = "YES" THEN GOTO 260$
360  PRINT"Wrong - the correct answer is "YES"
370  PRINT"Any array with more than 1 subscript must be DIMensioned."
380  PRINT"See part 2. ARRAYS."
390  GOTO 270$
400  PRINT"CORRECT"
410  I = 1
420  PRINT$
430  INPUT"press ENTER"$
440  GOSUB 400$
450  PRINT"Which of the following is valid"
460  PRINT$
470  PRINT M(2,1)$
480  PRINT"Wrong - & is illegal"
490  PRINT M(2,1)$
500  PRINT"Wrong - & is illegal"
510  PRINT$
520  PRINT$
530  INPUT"ENTER the letter opposite the correct answer"
540  PRINT$
550  IF A = " THEN GOTO 520$
560  PRINT"Wrong - the correct answer is "
**** Listing of Program 'TEST3' ****

2840 PRINT" Answer B has parens in the wrong place, C A D tr.
2850 PRINT" to use a string instead of a number for a pocket'
2860 PRINT" in the 'answer'.
2870 PRINT" See part 2. ARRAYS."
2880 GOTO 2910
2890 PRINT"CORRECT"
2900 D = 1
2910 PRINT
2920 INPUT 'Press ENTER if '
2930 GOSUB 4060
2940 PRINT"The CLEAR statement is used for clearing STRING space and'
2950 PRINT"ALWAYS must be used if your STRING use is greater than 100”
2960 PRINT"but doesn’t have to be used if your STRING will be LESS than'
2970 PRINT"100 characters."'
2980 PRINT
2990 PRINT"Is the above question TRUE or FALSE?"'
3000 PRINT
3010 INPUT 'Enter TRUE or FALSE: '
3020 PRINT
3030 IF T = "FALSE" GOTO 3090
3040 PRINT"CORRECT - 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 1. CLEAR."'
3080 GOTO 3110
3090 PRINT"CORRECT"
3100 D = 1
3110 PRINT
3120 INPUT 'Press ENTER: '
3130 GOSUB 4090
3140 FOR X = 1 TO 10
3150 Y = (X+1)
3160 NEXT X
3170 PRINT"You have finished the test. out of 10 possible correct answers"'
3180 PRINT"You scored "X"."'
3190 PRINT
3200 IF X = 10 THEN PRINT"YOU HAVE PASSED"
3210 GOSUB 5590
3220 IF X = 7 THEN GOTO 3090
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 1120

154
Do you want your score recorded on a permanent file?"  

IF YES THEN GOTO 5670  

PRINT"To record your score, we must open a file and put your name"  

PRINT"Therefore, surprisingly, we need your name. If your name is not unique among the students likely to take this test,"  

PRINT"then contact your test monitor for an identifying word that will make you unique. Then enter that word below."  

PRINT"If you have already entered a score previously, be sure to enter the same name you used before. Make all capitals!"  

INPUT"ENTER your word or name now:"  

OPEN"SCORE"  

READ Y  

IF EOF THEN GOTO 5570  

INPUTX  

INPUTY  

IF X = Y THEN GOTO 5750  

GOTO 5570  

CLOSE  

READ Y  

READ X  

FOR W = 1 TO X  

PRINT"*"  

PRINTY  

NEXT W  

PRINT"You are now qualified to go to LESSON 4."  

PRINT"If you want a homework assignment, select it now."  

INPUT"Do you want to see your homework (Y/N):"  

IF Y THEN GOTO 4140  

GOTO 4150  

GOTO 5560  

Y = Y
I.200 IF EDF(1) THEN CLOSE:X=X-1:SOTO 3800
I.250 INPUT#1. NS(1), S(1)
I.270 SOTO 3760
I.300 OPEN*".*.L."SCORE1"*
I.310 FOR W = 1 TO X
I.320 PRINT#1,W:S(W)
I.330 PRINT#1.1.S(W)
I.340 NEXT W
I.350 PRINT
I.360 PRINT"You may now take LESSON 4. You will be returned to the MENU"*
I.370 PRINT"from where you may go to LESSON 4 or quit."*
I.380 GOTO 4120
I.390 REM **
I.400 REM ** THIS ROUTINE LETS STUDENT KNOW WHAT AREA HE
I.410 REM ** OR SHE SHOULD REVIEW BEFORE GOING ON
I.420 REM **
I.430 IF "=10 THEN RETURN
I.440 PRINT"YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"*
I.450 PRINT
I.460 IF Q(1) = 0 THEN PRINT " part 1. KILL statement"*
I.470 IF Q(2) = 0 OR Q(3) = 0 OR Q(4) = 0 THEN PRINT " part 1. LOOPS"*
I.480 IF Q(5) = 0 OR Q(6) = 0 OR Q(7) = 0 OR Q(8) = 0 THEN PRINT" part 2. ARRAYS"*
I.490 IF Q(9) = 0 THEN PRINT " part 2. DIM statement"*
I.500 IF Q(10) = 0 THEN PRINT " part 2. CLEAR statement"*
I.510 PRINT
I.520 INPUitesse ENTER*T$
I.530 GOTO 4080
I.540 RETURN
I.550 REM **
I.560 REM this subroutine clears the screen'
I.570 REM **
I.580 FOR I = 1 TO 24
I.590 PRINT
I.600 NEXT I
I.610 RETURN
I.620 PRINT
I.630 INPUT'press ENTER to return to MENU':T$
I.640 RUN"MENU"
I.650 PRINT
I.660 INPUT'press ENTER to load your homework':T$
I.670 RUN "HW"
***** Listing of Program 'LESSON4' *****

1000 REM ** THIS PROGRAM STARTED ON 4 JUNE 1983
1010 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
1020 REM ** TITLE: LESSON 4A
1030 REM **
1040 REM **
1050 REM **
1060 REM **
1070 GOSUB 7020
1080 PRINT 'LESSON: BASIC 4 VERSION: 1 AUGUST 83
1090 PRINT
1100 PRINT 'TIME REQUIRED TO COMPLETE LESSON: About 1.5 hours'
1110 PRINT
1120 PRINT
1130 PRINT 'AUTHOR: Capt Danny J. Creagan'
1140 PRINT 'Air Force Institute of Technology'
1150 PRINT
1160 PRINT 'OBJECTIVE: To teach the student how to make the computer 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: Ti$
1240 CLEAR 200
1250 LL = 0 ; 'LL IS USED FOR HUMDR IN OPEN STATEMENT SECTION
1260 GOSUB 7020
1270 GOSUB 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: $'
1350 IF T$ = "C" THEN GOTO 7490
1360 IF T$ = "D" GOTO 7470
1370 IF T$ = "B" GOTO 1510
1380 IF T$ = "A" GOTO 1290
1390 GOSUB 1520
1400 GOSUB 1980
1410 GOSUB 2980
1420 GOSUB 4630
1430 GOSUB 9200
1440 GOSUB 7020
1450 PRINT 'You are now done with this lesson. If you wish to continue'
Listing of Program 'LESSON4' 07/10/83 - 01:56:17

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. Also enter a 0 to go to the MENU. ENTER a S to go to the second half.*"
1560 PRINT
1570 PRINT
1580 INPUT"What is your choice":N
1590 IF N = 0 GOTO 7470
1600 DM N GOSUB 1520 .1960 .2960 .6630 .7480
1610 GOTO 1510
1620 GOSUB 7020
1630 PRINT"Introduction"
1640 PRINT
1650 PRINT"In this lesson we will start learning how to communicate with"
1660 PRINT"our disk drives and printers. The first section deals with"
1670 PRINT"printing our programs on a printer. We can either print the"
1680 PRINT"output (ie the answer that our program calculated), or we can"
1690 PRINT"print our program listing to a printer."
1700 PRINT
1710 PRINT"The remaining sections will show us how to store data on a"
1720 PRINT"disk, so that we can save important calculations for future"
1730 PRINT"use."
1740 PRINT
1750 PRINT"Both sections are very important to the programmer and you"
1760 PRINT"will find yourself using them often."
1770 PRINT
1780 INPUT"Press ENTER":$n
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."
LISTING OF PROGRAM 'LESSON 4' 07/10/83 - 01:56:17

1920 PRINT "Remember, looking up the answers is not cheating, it's LEARNING."
1930 PRINT
1940 INPUT "press ENTER"; T$
1950 RETURN
1960 GOSUB 7020
1970 PRINT* LPRINT & LLIST*
1980 PRINT
1990 PRINT "LPRINT and LLIST allow you to output information to the line"
2000 PRINT "printer. They are extremely simple to use and they work."
2010 PRINT "almost exactly like the PRINT and LIST words. There are."
2020 PRINT "only a couple of things you need to keep in mind when you."
2030 PRINT "use them."
2040 PRINT
2050 PRINT "First, make sure the printer is hooked up and turned on.";
2060 PRINT* and, if you are using a Cromemco, make sure the "
2070 PRINT* printer is LINKed to your terminal, ask your"
2080 PRINT* operator how to use the LINK command."
2090 PRINT
2100 PRINT "Second, make sure there is enough paper in the printer."
2110 INPUT "press ENTER"; T$
2120 GOSUB 7020
2130 PRINT* LPRINT & LLIST*
2140 PRINT
2150 PRINT "To LPRINT a STRING to the printer, you must enclose it in"
2160 PRINT "quotes, just like the PRINT statement. When you LPRINT a"
2170 PRINT "numeric variable, you do not enclose it in quotes."
2180 PRINT "You will not see either on the screen. They will only print"
2190 PRINT "on the printer. Examples of valid LPRINT statements are:";
2200 PRINT
2210 PRINT* 10 LPRINT "CHRS(34)" "MONTHLY TRAINING REPORT" CHRS(34)
2220 PRINT
2230 PRINT* 10 A = 10
2240 PRINT* 20 LPRINT A
2250 PRINT
2260 INPUT "press ENTER"; T$
2270 GOSUB 7020
2280 PRINT* LPRINT & LLIST*
2290 PRINT
2300 PRINT* You may LPRINT TABs also, (just as you can PRINT TABs)"
2310 PRINT* However, the TAB function reacts differently on different"
2320 PRINT* machines. The general format for LPRINTing a TAB is:";
2330 PRINT
2340 PRINT* 10 LPRINT TAB(1) "CHRS(34)" "data" CHRS(34)
2350 PRINT
2360 PRINT*
**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
2480 PRINT LPRINT & LLIST"
2490 PRINT
2500 INPUT"press ENTER";T$
2510 GOSUB 7020
2520 PRINT"LLIST works just like LIST only it outputs to the printer"
2530 PRINT
2540 PRINT"It is normally used from the IMMEDIATE mode when you want"
2550 PRINT"to see your program lines on paper it is sometimes easier"
2560 PRINT"to find bugs in your program if you can see it on a piece"
2570 PRINT"of paper."
2580 PRINT
2590 PRINT P
2600 INPUT"press ENTER";T$
2610 GOSUB 7020
2620 PRINT"LLIST outputs data to the printer and to"
2630 PRINT"the screen."
2640 PRINT
2650 PRINT"Is the above sentence TRUE or FALSE?"
2660 PRINT
2670 INPUT"press ENTER";T$
2680 GOSUB 7020
2690 PRINT Which of the following commands will LLIST your entire
2700 PRINT"
2710 IF T$ = "S" THEN GOTO 2750
2720 PRINT"LLPRINT & LLIST only output to the printer"
2730 GOTO 2760
2740 PRINT"CORRECT"
2750 PRINT
2760 INPUT"press ENTER";T$
2770 GOSUB 7020
2780 PRINT"Which of the following commands will LLIST your entire"
2790 PRINT"program to the printer?"
2800 PRINT
2810 PRINT
2820 PRINT"A LLIST"
2830 PRINT"B LLIST ALL"
#### Listing of Program 'LESSON4' ####

07/10/83 - 01:56:17

```
2840 PRINT"C LIST "CHR$(34) '""ALL"CHR$(34)" (Ifn is your program name)"
2850 PRINT
2860 INPUT"ENTER the letter opposite the correct answer":T$  
2870 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 PRINT"Sequential Files Intro"
3000 PRINT
3010 PRINT"Remember, in the previous lessons, we learned that we could"
3020 PRINT"INPUT data into our programs, but that the data was not stored"
3030 PRINT"permanently. If we turned the machine off or otherwise ended"
3040 PRINT"our program, all the data that we ENTERed was lost. If we"
3050 PRINT"wanted to RUN the program again, we had to re-ENTER the data."
3060 PRINT
3070 PRINT"In the rest of this lesson, we will learn how to store our data"
3080 PRINT"that we ENTERed on a disk FILE. When we do that, we can always"
3090 PRINT"recall it for future use, and we won't have to keep entering"
3100 PRINT"the same information 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 PRINT"Sequential Files"
3170 PRINT
3180 PRINT"A disk file is an organized collection of data, such as a "
3190 PRINT"training record, or a mailing list. It is usually composed of "
3200 PRINT"just the data, and nothing else. Program statements or BASIC"  
3210 PRINT"words are normally not stored in file format."
3220 PRINT
3230 INPUT"press ENTER":T$
3240 GOSUB 7020
3250 PRINT"Sequential Files"
3260 PRINT
3270 PRINT
3280 PRINT"To transfer data from a BASIC program to a disk file, you must" 
3290 PRINT"create a BUFFER in memory. The data is first transferred to "
```

LISTING OF PROGRAM 'LESSON4' 07/10/81 - 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"n
3320 PRINT"are SEQUENTIAL files and DIRECT access files. We will only"n
3330 PRINT"explain SEQUENTIAL files in this lesson. If you need to "n
3340 PRINT"learn DIRECT access files, this lesson will still help you"n
3350 PRINT"because many of the commands are similar. Your BAsIC manual"n
3360 PRINT"will explain the differences DIRECT access is often called"n
3370 PRINT"RANDOM access by some manuals."
3380 PRINT
3390 INPUT"press ENTER":T$ 7020
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"n
3440 PRINT"access files do."
3450 PRINT
3460 PRINT"A TRUE*"
3470 PRINT"B FALSE*"
3480 PRINT"C I DON'T KNOW*"
3490 PRINT
3500 INPUT"ENTER the letter opposite the correct answer":T$ 7020
3510 PRINT
3520 IF T$ = "C" THEN GOSUB 7020:PRINT"Make a guess":PRINT:S0TO 3420
3530 IF T$ = "B" THEN GOTO 3570
3540 PRINT"WRONG - you ALWAYS have to create a BUFFER in memory*"n
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"n
3580 PRINT"next parts of the lesson.*"
3590 PRINT
3600 INPUT"press ENTER":T$ 7020
3610 GOSUB 7020
3620 PRINT* Sequential Files *
3630 PRINT
3640 PRINT"With a sequential file, you must access the data the same way"n
3650 PRINT"you wrote it to the disk. If you were to write the ages of "n
3660 PRINT"three people to the disk, using sequential access mode, you"n
3670 PRINT"could only read the last age you entered by reading ages one"n
3680 PRINT"and two first. They would be stored in a line, and the"n
3690 PRINT"computer would have to start with the first age and search"n
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"n
3720 PRINT"computer still does a good job of it. Your scores for the"n
3730 PRINT"previous tests were logged into a file using SEQUENTIAL access"n
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 INPUT 'C I wish to take the 5th'
3850 PRINT  
3860 INPUT 'ENTER the letter opposite the correct answer'; T$  
3870 PRINT  
3880 IF T$ = 'C' THEN GOSUB 7020: PRINT 'Have you always had a drinking problem?'; PRINT: PRINT 'Try again'; PRINT: GOTO 3780  
3890 IF T$ = 'B' THEN GOTO 3920  
3900 PRINT 'WRONG - the correct answer is B'
3910 GOTO 3920  
3920 PRINT 'CORRECT - good answer'
3930 PRINT  
3940 INPUT 'press ENTER'; T$  
3950 GOSUB 7020  
3960 PRINT 'Sequential Files Intro'
3970 PRINT  
3980 PRINT 'The statements and functions used with sequential files are: '
3990 PRINT  
4000 PRINT 'OPEN INPUT$ EOD'
4010 PRINT 'CLOSE INPUT$ CLOSE$'
4020 PRINT  
4030 PRINT 'We will be covering these words in greater detail in the'
4040 PRINT 'following sections. You don't have to memorize them now, just'
4050 PRINT 'lock them over. Generally, you would OPEN the file, either'
4060 PRINT 'INPUT your data, or PRINT it out to the disk, and then you'
4070 PRINT 'would CLOSE the file before going on.'
4080 PRINT  
4090 INPUT 'press ENTER'; T$  
4100 GOSUB 7020  
4110 GOSUB 7290  
4120 PRINT 'This is an example of how to create a SEQUENTIAL file, write'
4130 PRINT 'some data to it, and then CLOSE it. We will be examining this'
4140 PRINT 'program, and a companion program that will INPUT data from'
4150 PRINT 'the file we created, in the next sections. Generally speaking'
4160 PRINT 'line 10 OPENS the file (we'll explain more later), line 20'
4170 PRINT 'asks the operator to INPUT his/her name, and line 30 writes'
4180 PRINT 'the name out to a file called 'TEST'. Line 40 CLOSES the file.'
4190 PRINT  
4200 INPUT 'press ENTER'; T$
**** Listing of Program 'LESSON4' ****

```
4210 GOSUB 7020
4220 PRINT "Sequential access files are written to a disk in order, and"
4230 PRINT "you cannot access a piece of information from the middle of the"
4240 PRINT "file without searching through all the records for data pieces;"
4250 PRINT "from first to the one that has the record you want."
4260 PRINT
4270 PRINT "Is the above statement TRUE or FALSE?"
4280 PRINT
4290 PRINT "A TRUE"*  
4300 PRINT "B FALSE"
4310 PRINT "Uhhh... somewhere in-between?"
4320 PRINT
4330 INPUT "ENTER the letter opposite the correct answer": T$ 
4340 PRINT
4350 IF T$ = "A" GOTO 4410
4360 IF T$ = "B" THEN PRINT "Uhhhh..."
4370 PRINT "WRONG - This is a key concept, please go back and"
4380 PRINT "review this section before going on. You will be"
4390 PRINT "given the opportunity to review in a few moments"
4400 GOTO 4420
4410 PRINT "CORRECT - THAT WAS IMPORTANT TO UNDERSTAND"
4420 PRINT
4430 INPUT "press ENTER": T$ 
4440 GOSUB 7020
4450 PRINT "The two types of file modes, SEQUENTIAL and DIRECT."
4460 PRINT
4470 PRINT "Is the above statement TRUE or FALSE?"
4480 PRINT
4490 PRINT "A TRUE"*
4500 PRINT "B FALSE"
4510 PRINT
4520 INPUT "ENTER the letter opposite the correct answer": T$
4530 PRINT
4540 IF T$ = "A" GOTO 4570
4550 PRINT "WRONG - the two modes ARE called SEQUENTIAL and DIRECT"
4560 GOTO 4580
4570 PRINT "CORRECT"
4580 PRINT
4590 INPUT "press ENTER": T$
4600 GOSUB 7180
4610 IF T$ = "B" GOTO 4680
4620 RETURN
4630 GOSUB 7020
4640 PRINT "OPEN statement" 
4650 PRINT
4660 GOSUB 7130
```
4670 PRINT"In the above program, line 10 OPENS the file we wish to make."
4680 PRINT"Whenever you work with a file, you MUST OPEN it first, then"
4690 PRINT"manipulate the data, and then CLOSE it. If you try to write"
4700 PRINT"DATA to a disk without OPENing the file, you will get an ERROR."
4710 PRINT"message. (Also, don't OPEN a file that is already OPEN.)"
4720 PRINT
4730 PRINT
4740 INPUT"press ENTER";T$
4750 GOSUB 7200
4760 PRINT"Can you OPEN a file that is already OPEN?"
4770 PRINT
4780 INPUT"ENTER the letter opposite the correct answer";T$
4790 PRINT
4800 IF T$ ="B" THEN GOTO 4890
4810 PRINT"WROGlNG - the correct answer is B"
4820 GOTO 4870
4830 PRINT"CORRECT"
4840 INPUT"press ENTER";T$
4850 GOSUB 7100
4860 PRINT"To recap then, the format for the OPEN statement is:"
4870 PRINT" OPEN"CHR$(34)+"(mode"CHR$(34)+",buffer "CHR$(34)+",filename"CHR$(34)")"
4880 PRINT
4890 IF T$ ="B" THEN GOTO 5000
5000 PRINT"OPENING DATA to a disk is '0'. If you are"
5010 PRINT"OPENING data, the mode is '1'. You can have up to 3 buffers"
**** Listing of Program "LESSON4" ****

5120 PRINT"quotes."
5129 PRINT
5130 INPUT"press ENTER":T$
5140 GOSUB 7020
5150 PRINT"What are the two file nodes?"
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!"
5280 PRINT
5290 INPUT"press ENTER":T$
5300 GOSUB 7020
5310 PRINT* OPEN statement*
5320 PRINT
5330 GOSUB 7350
5340 PRINT"Here is an example of an OPEN statement (line 10) that opens"
5350 PRINT"a file for INPUT. Notice that the node 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 nodes '0' and '1'?"
5440 PRINT
5450 PRINT"A  '0' tells the computer that you are going to write to disk"
5460 PRINT"and '1' tells the computer you are going to input from disk"
5470 PRINT
5480 PRINT"B  '0' tells the computer that the files section is ON and"
5490 PRINT" '1' tells the computer that you want to interroGATE"
5500 PRINT
5510 PRINT"C  '0' tells the computer you want to Organize files and '1'"
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 'LESSON4' ****

5570 IF T$ = "C" THEN GOSUB 7020:LL = 1:PRINT"Starting countdown now....";PRINT:"PRESS ENTER for
LAUNCH back to the question":T$;GOTO 5420
5580 PRINT"WRONG - the correct answer is A"
5590 GOTO 5610
5600 PRINT"CORRECT"
5620 INPUT"press ENTER":T$
5630 IF LL = 1 THEN PRINT:PRINT"FIVE HOURS TO SHUTTLE LAUNCH .... COUNTDOWN CONTINUES":PRINT:"PRESS
ENTER":T$
5640 GOSUB 7020
5650 PRINT* OPEN Statement*
5660 PRINT
5670 GOSUB 7290
5680 PRINT"The buffer number can be any number between 1 and 3. If we"
5690 PRINT"use the buffer for one file, and later we OPEN another file in"
5700 PRINT"the same program, we cannot use the same buffer number. It"
5710 PRINT"MUST BE DIFFERENT. If there is more than one file OPEN at"
5720 PRINT"the same time, then they must be using different buffers!!!"
5730 PRINT
5740 INPUT"press ENTER":T$
5750 IF LL = 1 THEN PRINT:PRINT"FOUR HOURS TO SHUTTLE LAUNCH .... COUNTDOWN CONTINUES":PRINT:"PRESS
ENTER":T$
5760 GOSUB 7020
5770 PRINT"How many files with the same buffer number can we have OPEN?"
5780 PRINT"at the same time?"
5790 PRINT
5800 PRINT"A ONE B TWO C THREE D FOUR"
5810 PRINT
5820 INPUT"ENTER the letter opposite the correct answer":T$
5830 PRINT
5840 IF T$ = "A" THEN GOTO 5870
5850 PRINT"WRONG - the correct answer is A"
5860 GOTO 5880
5870 PRINT"CORRECT"
5880 PRINT
5890 INPUT"Press ENTER":T$
5900 GOSUB 7020
5910 GOSUB 7280
5920 PRINT"You must enclose the filename in quotes if it is a character"
5930 PRINT"string. However, the following is also legal:"n
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 ***"
**** Listing of Program 'LESSON4' ****

6000 PUSH
6010 PRINT "If you use a STRING variable for a filename, you do not enclose it in quotes. Look at the example above."
6020 PRINT
6030 INPUT "press ENTER";T$
6040 GOSUB 7020
6050 PRINT "Is the following statement valid?"
6060 PRINT
6070 PRINT "OPEN'CHRS(34)'"""CHRS(34)""""CHRS(34)""
6080 PRINT
6090 PRINT "A YES"
6100 PRINT "B NO"
6110 PRINT
6120 INPUT "ENTER the letter opposite the correct answer";T$
6130 IF T$ = "B" THEN GOTO 6130
6140 PRINT "WRONG - the correct answer is B. XX$ should not have quotes"
6150 GOTO 6120
6160 PRINT "CORRECT"
6170 PRINT
6180 INPUT "press ENTER";T$
6190 IF LL = 1 THEN PRINT "THREE HOURS TO LAUNCH .... COURICAN CONTINUES";PRINT;INPUT "PRESS ENTER";T$
6200 GOSUB 7020
6210 GOSUB 7280
6220 PRINT "When line 10 OPENs the file 'TEST', the computer searches".
6230 PRINT "available disk space to see if the file already exists. If it".
6240 PRINT "DOESN'T EXIST, THE COMPUTER WILL CREATE IT AUTOMATICALLY:"
6250 PRINT "this is only true for the INPUT side, if INPUT side, the file".
6260 PRINT "must have existed previously, or BASIC will print an ERROR"
6270 PRINT
6280 PRINT "REMEMBER ALSO, unless you've CLOSED a file that has previously".
6290 PRINT "been opened, you cannot use the buffer number again";
6300 PRINT
6310 INPUT "press ENTER";T$
6320 GOSUB 7290
6330 PRINT "If the file 'TEST' exists, then line 10 will re-open it, line".
6340 PRINT "it WILL WRITE OVER THE PREVIOUS DATA IN THE FILE, and line".
6350 PRINT "it WILL CLOSE the file. EVEN IF THE FILE WAS SEVERAL THOUSAND".
6360 PRINT "WORDS LONG, AFTER THE ABOVE PROGRAM IS RUN, IT WILL ONLY BE".
6370 PRINT "AS LONG AS THE NAME THAT WAS ENTERED IN LINE 20:"
6380 PRINT
6390 PRINT "Sequential files must be loaded into memory, manipulated and".
6400 PRINT "written back out in their entirety, you cannot just write".
6410 PRINT "a single record onto the front of the file that's on the disk."
**** Listing of Program 'LESSON4' ****

6490 INPUT 'press ENTER' + IT
6470 GOSUB 7020
6490 PRINT "Write in the statement that will OPEN a file for OUTPUT to disk."
6490 PRINT "Use buffer number 1. The filename is 'WIG'."
6500 PRINT
6510 PRINT "Do not use a line number (although you could normally), and do not use a number of blanks."
6520 PRINT
6530 LINE INPUT 'Enter your answer, a text string': TXT
6540 PRINT
6550 SE=OPEN"*CHRS(4)+"*CHRS(4)+".3"*CHRS(4)+"*CHRS(4)+"*CHRS(4)+"*CHRS(4)+"*CHRS(4)+"*CHRS(4)+"*CHRS(4)
6560 IF #=$A THEN GOTO 7020
6570 PRINT: "This section is difficult to visualize sometimes. How?"
6580 PRINT: "It is very important. You may wish to review it later."
6590 PRINT: "The correct answer is: 'G'"
6600 GOTO 7020
6610 PRINT: "The correct answer is: without a doubt you are a programmer."
6620 PRINT
6630 INPUT 'press ENTER':
6640 CALL 7020
6650 PRINT: "When a sequential file is OPENed, that formerly had data in it, the data is destroyed."
6660 PRINT: "What must you be sure to do?"
6670 PRINT
6680 PRINT: "Not write in the middle of the file unless it is absolutely 100% necessary to do so."
6690 PRINT: "Always INPUT all the data, manipulate it, then PRINT it back out."
6700 PRINT: "Nothing. You cannot OPEN a file that was previously created by PRINT. Write only on the END of the file."
6710 PRINT
6720 INPUT: "The letter opposite the correct answer is:"
6730 PRINT
6740 IF #=$A THEN GOTO 7020
6750 PRINT: "The correct answer is B".
6760 GOTO 7020
6770 PRINT: "Input: the correct answer is:"
6780 PRINT: "If L = 1 then PRINT; SHUTLE LAUNCH ABORTED .... COMPUTER MALFUNCTION BLAMED; PRINT: INPUT 'PRESS ENTER':"
6790 CALL 7020
6800 IF #=$A THEN GOTO 7020
6810 RETURN
6820 GOSUB 720
6830 RETURN
6840 GOSUB 710
**** Listing of Program LESSON4 ****  

10 PRINT "You have completed this portion of Lesson 4. If you wish,"
20 PRINT "to continue on to the next half of this lesson, enter a 'C'."
30 PRINT "If you wish to review this lesson again, enter an 'R'."
40 PRINT
50 INPUT "Enter your choice now!"; R$  
60 IF R$ = "C" THEN GOTO 160  
70 IF R$ = "R" THEN GOTO 460  
80 GOTO 10
90 REM **
100 REM ** clear screen subroutine
110 REM **
120 FOR T = 1 TO 24
130 PRINT
140 NEXT T
150 RETURN
160 REM ** This is the new subroutine
170 PRINT "LESSON 4"
180 PRINT
190 PRINT "This is the first part of a two part lesson. It is divided"  
200 PRINT into the following sections:"
210 PRINT
220 PRINT "1. Introduction 2. Sequential Files Intro."
230 PRINT "3. PRINT A LIST 4. OPENing files"
240 PRINT
250 PRINT
260 RETURN
270 REM ** This subroutine lets student review a section
280 GOSUB 100  
290 PRINT "Which do you wish to do?"
300 PRINT "1. Continue on"
310 PRINT "2. Review this section again"
320 PRINT
330 INPUT "Enter the letter opposite the correct answer and press ENTER":  
340 IF T = 1 OR T = 2 THEN RETURN
350 GOTO 200
360 REM ** This subroutine is for sequential intro example
370 PRINT: OPEN "CHARS.DAT" FOR INPUT AS ":CHR$ (1), MESSAGE "TEST CHR$ (1)
380 PRINT: INPUT "You in your name and press ENTER":CHR$ (1)
390 PRINT: PRINT CHR$ (1)
400 PRINT: CLOSE 1
410 GOTO 300
420 REM ** This subroutine is for the sequential files intro example
430 PRINT: OPEN "CHR$ (1).DAT" FOR INPUT AS ":CHR$ (1), MESSAGE "TEST CHR$ (1)
440 PRINT: INPUT "Your name":CHR$ (1)
***** Listing of Program "LESSON4" *****

7070 PRINT"I 0 PRINT M$
7080 PRINT"A CLOSE 
7090 PRINT
7100 RETURN
7110 REM ** This subroutine is for the OPEN statement example
7120 PRINT
7130 PRINT"10 M$ = CHR$(14)"""TEST"CHR$(14)
7140 PRINT"20 OPEN "CHR$(14)"""CHR$(14)""",1,M$
7150 PRINT
7160 RETURN
7170 RUN"MENU$
7180 PRINT
7190 PRINT"RE PRINT"Going to the second half of the lesson. Wait one second"
7200 RUN"LESSON4$
7210 GOSUB 7230
7220 PRINT"Going to TEST 4 - please stand."
7230 RUN"TEST4$
7240 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 = "C" GOTO 5740
IF T = "D" GOTO 7770
IF T = "F" GOTO 1259
IF T = "A" GOTO 1040
GOSUB 1030
GOSUB 1770
GOSUB 2590
GOSUB 4650
GOSUB 2290
GOSUB 1250

You are now done with this lesson. If you wish to continue.

Press ENTER to repeat a test. ENTER a 1. If you want to review. ENTER an R.

Enter your choice:

IF T = "T" GOTO 5730
IF T = "R" GOTO 5640
GOTO 115:
GOSUB 520:
GOSUB 520:

Please type in the number beside the area you wish.

Press ENTER to review 1 through 51 and then press ENTER - ENTER a 0 to go.

Press to the Menu.

Press what is your choice:

IF N = 0 GOTO 5740:
GOSUB 1250 (1770,3250,4150,5780)
GOTO 1250
GOSUB 520

CLOSE Statement

CLOSE buffer #...

The format for CLOSE is the BASIC word CLOSE plus an optional:
** Listing of Program 'LESSON4A' **

```
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:IT#"
1510 S$ = "OPEN" + CHR$(14) + "O" + CHR$(34) + "A" + CHR$(14) + "QUESTION" + CHR$(14)
1520 GOSUB 5290
1530 PRINT "is the following program valid (lines 20 & 30 are good)"
1540 PRINT
1550 PRINT "$#
1560 PRINT "$C
1570 PRINT "$D
1580 PRINT "$E
1590 PRINT "$F
1600 PRINT "$G
1610 PRINT
1620 PRINT "YES it is valid"
1630 PRINT "NO it will fail because line 50 OPENs a file already used"
1640 PRINT
1650 INPUT "ENTER the letter opposite the correct answer:IT#"
1660 PRINT
1670 IF $ = "A" THEN GOTO 1710
1680 PRINT "WRONG - file #2 was CLOSED by line 40, so buffer 3 could be"
1690 PRINT "used again in line 50."
1710 GOTO 1710
1713 PRINT "CORRECT"
1720 PRINT
1730 INPUT "press ENTER:IT#"
1740 GOSUB 5290
1750 IF $ = "$" GOTO 1760
1763 RETURN
1770 GOSUB 5290
1780 PRINT "ENTER"
1790 PRINT * 1
1800 PRINT
1810 GOSUB 5460
1820 PRINT "In line 20 we PRINT to the file buffer that we OPENed in line"
1830 PRINT "the variable $A. Notice that the number to the right of"
1840 PRINT "the PRINT corresponds to the buffer number. If the buffer"
1850 PRINT "was number 3, then, after we OPENed the file, we would"
1860 PRINT "PRINT "PRINT$D", "$ in line 20"
1870 PRINT
1870 PRINT
1877 INPUT "press ENTER:IT#"
1878 GOSUB 5290
1879 PRINT *
1880 PRINT
1890 PRINT "PRINT$D prints items to a sequential disk file. When you first-
```
Listing of Program 'LESSON4'

1920 PRINT "OPEN the file. the computer sets a pointer at the beginning"
1920 PRINT "when you tell the computer to PRINT something."
1940 PRINT "It starts writing data to the disk at the place where the"
1970 PRINT "pointer is. At the end of the PRINT# operation, the pointer"
1970 PRINT "advances, so values are written in sequence. PRINT# writes"
1970 PRINT "data to the disk almost exactly the way PRINT writes data to"
1970 PRINT "the screen (or LPRINT writes to the printer)."
1970 PRINT
2000 INPUT "PRESS ENTER:
2010 GOSUB 200
2020 PRINT"
2030 PRINT
2340 PRINT "Comas and semi-colons react the same way with PRINT# that they"
2350 PRINT "do with PRINT statements. If you were to write the following"
2360 PRINT "program:"
2370 PRINT
2380 PRINT 10 A=10.3
2390 PRINT 20 B=20.2
2400 PRINT
2410 PRINT "to a disk using PRINT#1.A,B (as opposed to PRINT#1,A,B). then"
2420 PRINT "you would put this on the disk: 10.3  20.2"
2430 PRINT
2440 PRINT "See the extra spaces. Those are 10 blanks that BASIC writes to"
2450 PRINT "the disk."
2460 PRINT
2470 INPUT "PRESS ENTER:
2480 GOSUB 200
2490 PRINT "PRINT#1,A,B"
2500 PRINT
2510 PRINT "The above command puts this on disk: 10.3  20.2"
2520 PRINT
2530 PRINT "If you use a semicolon, like this. PRINT#1,A,B then you get:"
2540 PRINT
2550 PRINT 10.3  20.2
2560 PRINT "PRINT" "There are only three spaces between the numbers. So to save"
2570 PRINT "space on the disk, you may want to write to disk using semi-"
2580 PRINT "colons instead of commas between your variables. Either way"
2590 PRINT "will work. It's just that the commas cause 10 extra blanks to"2590 PRINT "to the disk."
2600 PRINT
2610 INPUT "PRESS ENTER:
2620 GOSUB 200
2630 PRINT "what is wrong with the following program:"
2640 PRINT
2650 PRINT 10 OPEN CHR$(54)""""""".""CHR$(54)""""""""""NEWPROG""""CHR$(54)""""
### Listing of Program 'LESSON4A' **** 07/10/83 - 02:20:28

```
2380 PRINT"20 PRINT#1,14:22:55"
2390 PRINT"30 CLOSE 1"
2400 PRINT
2410 PRINT"A The mode is incorrect"
2420 PRINT"B The file buffer is incorrect"
2430 PRINT"C You cannot use semicolons between numbers in a PRINT#"
2440 PRINT"D Nothing"
2450 PRINT
2460 INPUT"Enter the letter opposite the correct answer":T$
2470 PRINT
2480 IF T$="A" GOTO 2520
2490 PRINT"Wrong - The correct answer is A - the mode is incorrect for"
2500 PRINT"PRINT#"
2510 GOTO 2530
2520 PRINT"CORRECT - way to go!"
2530 PRINT
2540 INPUT\"press ENTER\"":T$
2550 GOSUB 536
2560 IF T$="B" GOTO 1770
2570 RETURN
2580 GOSUB 5200
2590 PRINT\"INPUT$ & EOF\" INPUT$EOF
2600 PRINT
2610 PRINT\"INPUT$ is similar to the INPUT word that we learned earlier\"
2620 PRINT\"only it inputs data from a disk that previously had data\"
2630 PRINT\"printed to it.\"
2640 PRINT
2650 GOSUB 5570
2660 PRINT\"The format for the INPUT$ statement is similar to the PRINT#\"
2670 PRINT\"statement. It is: INPUT$ (buffer$) (variable1) (var2) etc\"
2680 PRINT
2690 INPUT\"press ENTER\"
2700 GOSUB 5200
2710 PRINT\"INPUT$\"
2720 PRINT
2730 PRINT\"INPUT$ inputs data from a sequential disk file and stores the\"
2740 PRINT\"data in a variable. INPUT$ doesn't care how data was placed on\"
2750 PRINT\"the disk. It could have been put there with one PRINT#\"
2760 PRINT\"or twenty PRINT# statements. WHAT MATTERS TO INPUT$ IS HOW THE\"
2770 PRINT\"DATA IS TERMINATED ON THE DISK, AND WHAT KIND OF DATA IT IS\"
2780 PRINT\"INPUT$ TERMINATING.\"
2790 PRINT
2800 PRINT
2810 INPUT\"press ENTER\"
2820 GOSUB 5200
2830 PRINT\"Does the INPUT$ statement check to see how the data was\"
2840 PRINT\"placed on the disk, or does it check to see how the data is\"
```

177
2840 PRINT "terminated"
2850 PRINT
2860 PRINT "A: It only checks to see how it was terminated - it doesn't"
2870 PRINT "care how the data got there."
2880 PRINT
2890 PRINT "B: It checks to see how the data was placed. It makes a "
2900 PRINT "difference how many PRINT$ statements were used."
2910 PRINT
2920 PRINT "C: Now this question is easy. Just give me a second and I'll"
2930 PRINT "think of the answer. Hmm, let's see... No, don't tell me.."
2940 PRINT
2950 INPUT "Enter the letter opposite the correct answer": T$
2960 PRINT
2970 IF T$ = "A" THEN GOTO 3010
2980 IF T$ = "C" THEN PRINT "Times up!" .. because you took so much time, you get the answer "
2990 PRINT "Wrong - the answer is A"
3000 GOTO 3020
3010 PRINT "Correct - good job"
3020 PRINT
3030 INPUT "Press ENTER": T$
3040 GOSUB 5200
3050 PRINT "INPUT$ & EOF"
3060 PRINT
3070 PRINT "If we are inputting STRING data (our variable is a"
3080 PRINT "STRING such as INPUT1.N81, INPUT$ starts putting data into"
3090 PRINT "the variable starting with the first NON-SPACE it encounters"
3100 PRINT "in the file, and ending when it encounters a carriage return or"
3110 PRINT "a comma, or EOF marker (more about EOF later)."
3120 PRINT
3130 PRINT "If the variable is numeric, then INPUT$ fills the variable with"
3140 PRINT "the first character that is not a space or carriage return,"
3150 PRINT "and stops when it encounters another space, comma, or carriage"
3160 PRINT "return or EOF marker."
3170 PRINT
3180 INPUT "Press ENTER": T$
3190 GOSUB 5200
3200 PRINT "INPUT$ and EOF"
3210 PRINT
3220 PRINT "Here is an important concept to understand about how INPUT$"
3230 PRINT "works when you use strings. IF YOU PRINT A STRING TO DISK, AND"
3240 PRINT "YOU PRINT A NUMBER WITH IT, IT WILL LOOK LIKE THIS ON DISK:";
3250 PRINT
3260 PRINT "STRING DATA HERE WITH NUMBER FOLLOWING "1001
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

220 PRINT"ON THE LINE, (INCLUDING THE NUMBER) BECAUSE A STRING INPUT DOESN'T"
230 PRINT"RECOGNIZE SPACES AS TERMINATORS, IT ONLY RECOGNIZES CARRIAGE"
240 PRINT"RETURNS AND COMMAS:""
250 PRINT
260 PRINT"why is this important? Because you wrote the data to disk with"
270 PRINT"TWO variables, and you read it back with only ONE. If you tried"
280 PRINT"to read the number after you read in the string, you would not"
290 PRINT"find it":"
300 PRINT
310 INPUT"press ENTER":$*
320 GOSUB 5200
330 PRINT"the solution to the problem of how to write STRING data and"
340 PRINT"NUMERIC data to disk, is to separate them with carriage returns"
350 PRINT"or ENTERS). That way you can read them back with INPUT"*
360 PRINT"statement in the same way you wrote it. For example, if you"
370 PRINT"write a string and a number to disk, do it this way:"*
380 PRINT
390 PRINT"..... program assumes file opened correctly....."
400 PRINT"20 PRINT"CHR$:34)*PRINT "THE STRING WITH ONE LINE and the number with another."CHR$:34"
410 PRINT"30 PRINT$1,1001;"
420 PRINT
430 PRINT"And when you read it back, use two separate statements.""
440 PRINT
450 PRINT"10 INPUT#1,A$*
460 PRINT"20 INPUT#1,N$"
470 PRINT
480 PRINT"This will solve the problem of mixing strings and numbers on disk.""
490 PRINT
500 INPUT"press ENTER":$*
510 GOSUB 5200
520 GOSUB 5300
530 PRINT"If we were to run this program, and the name that was in the"
540 PRINT"first record in the file was 'DANNY JOE', line 20 would start"
550 PRINT"with the first byte of the first record that was not a space"
560 PRINT"or a carriage return, and load N$ with it, it would continue"
570 PRINT"loading N$ until it encountered either a carriage return or a"
580 PRINT"comma. If the data contained a quoted character string, then"
590 PRINT"all the data between the quotes would be stuffed into N$'
600 PRINT"(unless a comma or carriage return were encountered).'"
610 PRINT
620 INPUT"press ENTER":$*
630 GOSUB 5200
640 GOSUB 5300
650 PRINT"Eventually, and very quickly, the string would be loaded with"
660 PRINT"the characters 'DANNY JOE', and line 20 would print them out.'"
670 PRINT
680 PRINT
310 PRINT "But what would happen if there was nothing in the file?"
320 PRINT "The INPUT# statement in line 20 would encounter the EOF marker."
330 PRINT "And an ERROR would be returned saying that the computer tried"
340 PRINT "to input data that wasn't there."
350 PRINT
360 PRINT "There is a way to test to see if the file is at the end or if it is empty."
370 PRINT "Press ENTER to see what it is:"
380 GOSUB 5200
390 PRINT
400 PRINT "The very first thing a computer does when it OPENs a file, is"'
410 PRINT "to put a marker on the end of it. The marker is called an EOF"'
420 PRINT "marker. If we wanted to find out if the end of a file had been"
430 PRINT "reached, or if the file was empty, we would test it like this:"'
440 PRINT
450 INPUT "Press ENTER": T$'
460 GOSUB 5200
470 PRINT
480 PRINT "The first thing a computer does when it OPENs a file, is to put "'
490 PRINT "the EOF marker in place. If we had printed a thousand and one "'
500 PRINT "names into the file "TEST"", the EOF marker would put the EOF "'
510 PRINT "marker in place # 1001, and="'
520 PRINT "when we used the EOF(buffer #) test after name # 1001, control"
530 PRINT "would pass to line 60. All the names would have been printed!"
540 PRINT "Notice that we never re-OPEN a file if we haven't CLOSED it."
550 PRINT "If we tried to re-OPEN an already OPEN "file, we'd get an ERROR."
560 PRINT
570 INPUT "Press ENTER": T$'
580 GOSUB 5200
590 PRINT
600 PRINT "The format for the EOF statement is:"
610 PRINT
620 PRINT "EOF(buffer #)"'
630 PRINT
640 PRINT "Where buffer number corresponds to the buffer number of the "'
650 PRINT "file you are testing. "'
660 PRINT
670 PRINT "Remember to only use the EOF test on a file that is OPEN!"
680 PRINT
690 INPUT "Press ENTER": T$'
700 GOSUB 5200
710 PRINT "Suppose we have a file called "TEST" that is full of numeric "'
720 PRINT "data. What is wrong with the following program if we were "'
730 PRINT "trying to print the file out to the screen?"
740 PRINT
750 OPEN "CHR$ :A:"'
760 PRINT "CHR$(34):.1,"'
770 PRINT "CHR$(34):"'
780 PRINT "CHR$(34):"
Listing of Program 'LESSON4A'  

07/10/83 - 02:20:28

4220 PRINT"20 IF EOF(1) THEN STOP"
4230 PRINT"0 INPUT#1,N"  
4240 PRINT"40 PRINT N*"
4250 PRINT"50 GOTO 10*"
4260 PRINT"A Nothing*"
4270 PRINT"B If 'TEST' is empty, the EOF check won't catch it"  
4280 PRINT"C Line 50 should be GOTO 20*"
4290 PRINT"D The file mode is incorrect*"
4300 PRINT"E Line 50 should be SOTO 10*"
4310 PRINT"F Nothing*"
4320 INPUT"ENTER the letter opposite the correct answer*";T$I$
4330 IF T$ = "C" THEN GOTO 4380
4340 IF T$ = "D" 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"ENTER the letter opposite the correct answer*";T$I$
4410 GOSUB 52000
4420 PRINT"What is wrong with this program?*"
4430 PRINT  
4440 PRINT"10 OPEN"CHR$(34)*"CHRS$(34)*" LINE"CHR$(34)*"TEST"CHR$(34)*"  
4450 PRINT"20 IF EOF(2) THEN STOP*"
4460 PRINT"30 INPUT#1,N*"
4470 PRINT"40 PRINT N*$*"
4480 PRINT"50 GOTO 20*"
4490 PRINT  
4500 PRINT"A The wrong buffer number is used*"
4510 PRINT"B Nothing*"
4520 PRINT"C There is no END statement*"
4530 PRINT  
4540 INPUT"ENTER the letter opposite the correct answer*";T$I$
4550 PRINT  
4560 IF T$ = "A" THEN GOTO 4590
4570 PRINT"WRONG the correct answer is A*"
4580 GOTO 4600
4590 PRINT"CORRECT*"
4600 PRINT  
4610 INPUT"press ENTER*";T$I$
4620 GOSUB 5200
4630 IF T$ = "B" GOTO 2500
4640 RETURN
4650 GOSUB 5200
4660 PRINT*  
4670 PRINT

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 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.

```
4680 PRINT "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."
4710 PRINT "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."
4780 PRINT "On the following pages, there is a program that uses what we have learned in this lesson. Study it carefully."
```

```
4810 INPUT "Press ENTER"$;T$ 5200
4830 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." 5300
4880 PRINT "While you are studying the program, think about how you would do it without changing the filename."
4900 PRINT "HINT: the program would have to read the data into an array, add new data onto the array, and finally write the whole array back out to the old file."
4940 PRINT "Note how the comments are inserted. Look up why this is legal."
4970 PRINT "in your BASIC mania. You may be surprised!"
4970 PRINT
```

```
5000 INPUT "Press ENTER"$;T$ 5200
5010 PRINT "CLEAR 50:";"Clears string space"
5020 PRINT "OPEN"CHR$(34)"1"CHR$(34)"O"CHR$(34)"5"CHR$(34)"2"CHR$(34)"4"CHR$(34)"6"CHR$(34)"8"CHR$(34)"10"CHR$(34)"12O:"CHR$(34)"14"CHR$(34)"16"CHR$(34)"18":"Opens the INPUT file"
5030 PRINT "OPEN"CHR$(34)"1"CHR$(34)"O"CHR$(34)"5"CHR$(34)"2"CHR$(34)"4"CHR$(34)"6"CHR$(34)"8"CHR$(34)"10"CHR$(34)"12O:"CHR$(34)"14"CHR$(34)"16"CHR$(34)"18":"Opens the OUTPUT file"
5040 PRINT "IF EOF(1) GOTO 120:";"Checks for end of file in file #1"
5050 PRINT "INPUT#2.";"INPUTs data from file #1"
5060 PRINT "PRINT#2.";"PRINTs data to the new file"
5070 PRINT "GOTO 60:";"Goes back for more from file #1"
5080 PRINT "GOTO 60:";"Goes back for more from file #1"
5090 IF D$ = "CHR$(34)"999"CHR$(34)"100:""Checks for end of file in file #2"
5110 PRINT "GOTO 120:";"Updates new file with your data"
5120 PRINT "GOTO 120:";"Goes back until line 110 sees "999"
5130 PRINT "CLOSE:";"Closes files"
```
***** Listing of Program 'LESSON4A' *****

510 PRINT
5110 INPUT "We assume a legal input file previously existed...press ENTER": T$
5120 GOSUB 5130
5130 IF T$="S" GOTO 4650
5140 RETURN
5150 REM ** This subroutine clears the screen on any terminal
5200 FOR I = 1 TO 24
5210 PRINT
5220 NEXT I
5230 RETURN
5240 REM ** This is the menu subroutine
5250 PRINT
5260 RETURN
5270 PRINT
5280 PRINT "This is the second part of a two part lesson. It is divided"
5290 PRINT "into the following sections:*"
5300 PRINT
5310 PRINT "1) CLOSE"
5320 PRINT "2) PRINT#"
5330 PRINT "3) INPUT!EOF"
5340 PRINT "4) SUMMARY"
5350 PRINT "5) TEST"
5360 RETURN
5370 REM ** This subroutine lets student review a section
5380 GOSUB 5370
5390 PRINT "Which do you wish to do?"
5400 PRINT
5410 PRINT "A Continue on"
5420 PRINT "B Review this section again"
5430 PRINT
5440 INPUT "Press the letter opposite the correct answer and press ENTER": T$
5450 IF T$ = "A" OR T$ = "B" THEN RETURN
5460 GOTO 5450
5470 REM ** this subroutine is for sequential intro. example
5480 PRINT "OPEN *CHR$134)"CHR$134)".CHR$134)".CHR$134)"TEST*CHR$134)
5490 PRINT "2) INPUT*CHR$134)"Type in your name and press ENTER*CHR$134)"T$
5500 PRINT "3) PRINT#"
5510 PRINT "4) CLoSE 1"
5520 PRINT
5530 PRINT
5540 RETURN
5550 REM ** This subroutine is for the sequential files intro example
5560 PRINT "2) OPEN *CHR$134)"CHR$134)".CHR$134)"TEST*CHR$134)
5570 PRINT "3) INPUT#"
5580 PRINT "4) PRINT "FILE"
5590 PRINT "5) CLoSE 1"
5600 PRINT
5610 RETURN
5620 REM ** This subroutine is for the OPEN statements example
**** Listing of Program 'LESSON4A' ****

5600 PRINT
5610 PRINT 10 AS = "CHR$ (34)" * "CHR$ (34)"
5620 PRINT 20 OPEN "CHR$ (34)" * "CHR$ (34)"*, 1, .AS*
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, .AS*
5680 PRINT 40: PRINT .AS*
5690 PRINT 50: GOTO 20
5700 PRINT 60: PRINT "CHR$ (34)" * "END OF FILE ENCOUNTERED" * "CHR$ (34)"
5710 PRINT 70: CLOSE 1
5720 PRINT "...........more of the program or an END statement"  
5730 RETURN
5740 RUN "MENU"
5750 PRINT
5760 PRINT "GOING TO THE FIRST PART - WAIT ONE MOMENT"
5770 RUN "LESSON4"
5780 GOSUB 6200
5790 PRINT "GOING TO TEST 4 - Please standby"
5800 RUN "TEST4"
5810 END
***** Listing of Program 'TEST4' *****

1000 REM **
1010 REM ** LESSON: TEST4  VERSION: 1 AUB 83
1020 REM ** AUTHOR: CAPT DAN CREAGAN
1030 REM ** AIR FORCE INSTIUTE OF TECHNOLOGY
1040 REM **
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 ** U$(X) = ARRAY TO KEEP TRACK OF NUMBER OF CORRECT ANSWERS. IF AN ARRAY ELEMENT EQUALS 1, THE ANSWER WAS CORRECT
1100 REM **
1110 CLEAR 3000
1120 GOSUB 4400
1130 DIM N$(1000)
1140 DIM U$(10)
1150 DIM S$(1000)
1160 PRINT"FINAL TEST (lesson 4)"
1170 PRINT
1180 PRINT*"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 many things together for you. GOOD LUCK"
1190 PRINT
1200 INPUT"press ENTER to continue":T$ *17
1210 GOSUB 4400
1220 PRINT*"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."
1230 PRINT
1240 PRINT*"Is the above paragraph TRUE or FALSE?"
1250 PRINT*
1260 PRINT*A TRUE"
1270 PRINT*B FALSE"
1280 GOSUB 4470
1290 PRINT"" part i. LPRINT"

185
***** Listing of Program "TEST" ***** 07/10/82 - 02:53:37

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 "PRINT "CHR$(141"(filename)"CHR$(34)
1550 PRINT"B LIST "CHR$(34)(filename)"CHR$(34)
1560 PRINT"C LLIST"
1570 PRINT"D None of the above"
1580 PRINT
1590 GOSUB 4470
1600 PRINT
1610 IF T$ = "C" THEN GOTO 1670
1620 PRINT"WRONG - the correct answer is C"
1630 PRINT"Answer A would print the word 'filename' to the"
1640 PRINT"printer, answer B would cause a syntax error in BASIC."
1650 PRINT"See part 1, LLIST."
1660 GOTO 1690
1670 PRINT"CORRECT"
1680 Q(2) = 1
1690 PRINT
1700 INPUT"press ENTER":T$
1710 GOSUB 4400
1720 PRINT"You may access the middle record of a sequential file without"
1730 PRINT"reading in the records that are in front of it."
1740 PRINT
1750 PRINT"Is the above statement TRUE or FALSE"
1760 PRINT
1770 PRINT"A TRUE"
1780 PRINT"B FALSE"
1790 PRINT
1800 GOSUB 4470
1810 PRINT
1820 IF T$ = "B" THEN GOTO 1880
1830 PRINT"WRONG - the correct answer is B"
1840 PRINT"The only way to access a record in the middle of the"
1850 PRINT"file is to read the records in front of it."
1860 PRINT"See part 1, Sequential Files."
1870 GOTO 1900
1880 PRINT"CORRECT"
1890 Q(1) = 1
1900 PRINT
1910 INPUT"press ENTER":T$

186
1920 GOSUB 4400
1930 PRINT "To transfer data from your file to a disk, the computer must"
1940 PRINT "first send the data to a buffer where it is processed, the same"
1950 PRINT "is true for transferring data from the disk back to the"
1960 PRINT "computer."
1970 PRINT
1980 PRINT "Is the above paragraph TRUE or FALSE?"
1990 PRINT
2000 PRINT "A TRUE"
2010 PRINT "B FALSE"
2020 PRINT
2030 GOSUB 4470
2040 PRINT
2050 IF TI$ = "A" THEN GOTO 2110
2060 PRINT "WRONG - the correct answer is A"
2070 PRINT "Buffers are used for all data transfer between the"
2080 PRINT "computer and the disks."
2090 PRINT "See part 1. Sequential files, and part 2. OPEN & CLOSE"
2100 GOTO 2130
2110 PRINT "CORRECT"
2120 INPUT "press ENTER":TI$
2130 GOSUB 4400
2140 PRINT
2150 PRINT "You wish to access a previously created disk file. you will"
2160 PRINT "read in the data and use it to make an important financial"
2170 PRINT "decision. Which of the following statements will open the file"
2180 PRINT "and read the data into buffer 5. The filename is B52DAT"
2190 PRINT
2200 PRINT
2210 PRINT "A OPEN'CHR$(34)'0'CHR$(34)'I,'CHR$(34)'B52DAT'CHR$(34)"
2220 PRINT "B OPEN'CHR$(34)'1,5,'CHR$(34)'B52DAT'CHR$(34)"
2230 PRINT "C OPEN'CHR$(34)'1,'CHR$(34)'B52DAT'CHR$(34)"
2240 PRINT "D None of the above"
2250 PRINT "CORRECT"
2260 PRINT
2270 GOSUB 4470
2280 PRINT
2290 IF TI$ = "E" THEN GOTO 2380
2300 PRINT "WRONG - the correct answer is E"
2310 PRINT "the correct way is OPEN'CHR$(34)'1,'CHR$(34)'B52DAT'CHR$(34)
2320 PRINT "Answer A tries to open the file for output. answer B"
2330 PRINT "misplaces the 'I' and buffer number. answer C doesn't"
2340 PRINT "have quotes around the filename, and answer D should"
2350 PRINT "have quotes only around the 'I' and the filename."
2360 PRINT "See part 2. OPEN"
2370 GOTO 2400
***** Listing of Program 'TEST4' *****

2380 PRINT"CORRECT"
2390 Q(i) = 1
2400 PRINT
2410 INPUT"press ENTER":I$  
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)".I.CHR$(34)"TEST"CHR$(34)
2470 PRINT"20 IF EOF(i) THEN END"
2480 PRINT"30 INPUT#1,A$"
2490 PRINT"40 PRINT AS"
2500 PRINT".O0"
2510 PRINT
2520 PRINT"A Nothing"
2530 PRINT"B A 'File already open' ERROR will be generated"
2540 PRINT"C The wrong file mode is used"
2550 PRINT"D Line 20 is illegal"
2560 PRINT
2570 PRINT
2580 GOSUB 4470
2590 PRINT
2600 IF Ts = "B" THEN GOTO 2650
2610 PRINT"Wrong - the correct answer is B"  
2620 PRINT"Line 50 should be 'GOTO 20' to fix the problem."  
2630 PRINT"See part 2, OPEN and CLOSE"
2640 GOTO 2670
2650 PRINT"CORRECT"
2660 Q(i) = 1
2670 PRINT
2680 INPUT"press ENTER":I$  
2690 GOSUB 4400
2700 PRINT"The CLOSE word may be used without a buffer number; however,"
2710 PRINT"when you do that, only the most recent file is CLOSED."  
2720 PRINT
2730 PRINT"Is the above statement TRUE or FALSE?"
2740 PRINT
2750 PRINT"A TRUE"
2760 PRINT"B FALSE"
2770 PRINT
2780 GOSUB 4470
2790 PRINT
2800 IF Ts = "B" THEN GOTO 2850
2810 PRINT"Wrong - the correct answer is B"  
2820 PRINT"When the CLOSE word is used without a number, ALL"
2830 PRINT"the previously opened files are closed."
**** Listing of Program 'TEST4' ****

2340 PRINT" See part 2. CLOSE."
2350 GOTO 2890
2360 PRINT"CORRECT"
2370 Q(7) = 1
2380 PRINT
2390 INPUT"press ENTER":I$
2400 GOSUB 4400
2410 PRINT"What is wrong with the following program?"
2420 PRINT
2430 PRINT"OPEN"&CHR$(34)""&CHR$(34)&".1."&CHR$(34)""&CHR$(43)"#""&CHR$(34)"
2440 PRINT":20 PRINT#1,10;20:30"
2450 PRINT"50 CLOSE"
2460 PRINT
2470 PRINT
2480 PRINT"A Nothing"
2490 PRINT"B Wrong file mode is used"
2500 PRINT"C The CLOSE statement is invalid"
2510 PRINT"D The PRINT#1 statement should be INPUT#"1"
2520 PRINT
2530 GOSUB 4470
2540 PRINT
2550 IF $ = "A" THEN GOTO 3090
2560 PRINT"Wrong - the correct answer is A"
2570 PRINT"See part 2. OPEN and CLOSE"
2580 GOTO 3110
2590 PRINT"CORRECT"
2600 Q(6) = 1
2610 PRINT
2620 INPUT"press ENTER":I$
2630 GOSUB 4400
2640 PRINT"You have opened a file correctly and you now want to read in "
2650 PRINT"the data from it. Type in the command you would use."
2660 PRINT"Use line number 10. a space. and the command. Use buffer #8"
2670 PRINT"and read the data into variable AS"
2680 PRINT
2690 IF $ = "A" THEN GOTO 3250
2700 PRINT"WRONG - the correct answer is: 10 INPUT#8,AS"
2710 PRINT"See part 2. PRINT#, and INPUT#"
2720 GOTO 3270
2730 PRINT"CORRECT"
2740 GOTO 3110
2750 PRINT"CORRECT"
2760 Q(9) = 1
2770 PRINT
2780 INPUT"press ENTER":I$
2790 GOSUB 4400
*** Listing of Program 'TEST4' ***

5100 PRINT "What is wrong with the following program?"
5210 PRINT
5220 PRINT "10 IF EOF(1) THEN GOTO 40"
5230 PRINT "20 INPUT A,B"
5240 PRINT "30 GOTO 10"
5250 PRINT "40 CLOSE"
5260 PRINT "$ END"
5270 PRINT
5280 PRINT "Nothing"
5290 PRINT "The file wasn't properly opened"
5300 PRINT "Nothing will happen with the data"
5310 PRINT "The END statement is not needed"
5320 PRINT
5330 GOSUB 4470
5340 PRINT
5350 IF T$ = "9" THEN GOTO 3500
5360 PRINT "Wrong - the correct answer is 9"
5370 PRINT "The file should be opened before an EOF check is done."
5380 PRINT "See part 2"
5390 GOTO 3520
5400 PRINT "CORRECT"
5410 GOTO 10
5420 FOR Y = 1 TO 10
5430 T$ = Y$(11)
5440 NEXT X
5450 PRINT "You have finished the test. out of 10 possible correct answers:"
5460 PRINT "You scored "Y$"."
5470 PRINT
5480 IF Y > 9 THEN PRINT "YOU HAVE PASSED"
5490 GOSUB 4250
5500 IF Y < 7 THEN GOTO 3710
5510 PRINT "YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
5520 PRINT
5530 GOSUB 4400
5540 PRINT "YOU SHOULD RETAKE LESSON 4"
5550 PRINT
5560 PRINT "You will be returned to the Menu."
5570 PRINT
5580 GOTO 4490
5590 PRINT
5600 PRINT "Do you want your score recorded on a permanent file?"
5610 PRINT
5620 PRINT "YES" 
5630 PRINT "$ NO"
**** Listing of Program 'TEST4' ****

2260 PRINT
2270 INPUT "Which?": T$
2280 IF T$ = "9" THEN GOTO 4030
2290 GOSUB 4400
2300 PRINT "To record your score, we must open a file and put your name."
2310 PRINT "in it. Therefore, surprisingly, we need your name. In your"
2320 PRINT "name is not unique among the students for an identifying word that"
2330 PRINT "please contact your test monitor for an identifying word that"
2340 PRINT "will make you unique. Then enter that word below."
2350 PRINT
2360 PRINT "If you have already entered a score previously, be sure to"
2370 PRINT "enter the same name you used before. (use all capitals!)
2380 PRINT
2390 INPUT "Enter your word or name now": T$
2400 OPEN "", "FILE", "SCORE4"
2410 X = 0
2420 IF EOF(1) THEN GOTO 4980
2430 X = X + 1
2440 INPUT ".WS: X$"
2450 INPUT ".WS: X$"
2460 IF X$ = T$ THEN GOTO 4140
2470 GOTO 4980
2480 CLOSE
2490 X = X + 1
2500 X$ = T$
2510 BIT = X
2520 OPEN "", "FILE", "SCORE4"
2530 FOR W = 1 TO X
2540 PRINT ".WS: X$"
2550 PRINT ".WS: X$"
2560 NEXT W
2570 PRINT
2580 INPUT "Are you now qualified to go to LESSON 5?": T$
2590 PRINT
2600 PRINT
2610 INPUT "Do you wish a homework assignment (Y/N)?": T$
2620 IF T$ = "N" THEN GOTO 4310
2630 GOTO 4930
2640 X = X
2650 IF EOF(1) THEN CLOSE;GOTO 4190
2660 X = X + 1
2670 INPUT ".WS: X$"
2680 GOTO 4190
2690 OPEN "", "FILE", "SCORE4"
2700 F29 X = 1 TO X
**Listing of Program 'TEST4'**

4210 PRINT#1, "SIEM";
4220 PRINT#1, "SWI";
4230 NEXT M
4240 GOTO 4200
4250 IF # = 10 THEN RETURN
4260 PRINT "YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:"
4270 PRINT
4280 IF Q(1) = 0 THEN PRINT "LPRINT"
4290 IF S(2) = 0 THEN PRINT "LPRINT"
4300 IF Q(1) = 0 OR Q(4) = 0 THEN PRINT "Sequential Files"
4310 IF Q(5) = 0 OR Q(6) = 0 OR Q(7) = 0 THEN PRINT "OPEN and CLOSE"
4320 IF Q(8) = 0 OR Q(9) = 0 OR Q(10) = 0 THEN PRINT "PRINT", INPUT", and EOF"
4330 PRINT
4340 INPUT press ENTER to press INPUT
4350 GOTO 4400
4360 RETURN
4370 REM **
4380 REM ** this subroutine clears the screen
4390 REM **
4400 FOR I = 1 TO 24
4410 PRINT
4420 NEXT I
4430 RETURN
4440 REM **
4450 REM ** this subroutine is for the response section
4460 REM
4470 INPUT "ENTER the letter opposite the correct answer":
4480 RETURN
4490 PRINT
4500 INPUT press ENTER to return to MENU":
4510 RUN "MENU"
4520 CLOSE
4530 OPEN "\", "SCREA"*
4540 IF EOF = 1 THEN STOP
4550 INPUT#1, A.M.N
4560 PRINT A.M.N
4570 GOTO 4540
4580 RUN "444"
*.* Listing of Program 'LESSON5' *.*

00 REM ** THIS PROGRAM STARTED ON 1 JUN 93
01 REM ** AUTHOR: CAPTAIN DAMM J. CREAGAN
02 REM ** TITLE: LESSON S
03 REM **
04 REM **
05 REM **
06 REM **
07 REM **
08 REM **
09 REM **
10 REM **
11 REM **
12 REM **
13 REM **
14 REM **
15 D SUB 21000
16 PRINT 'LO LESSON: BASIC S VERSION: 1 AUGUST 93
17 PRINT
18 PRINT 'TIME REQUIRED TO COMPLETE LESSON: About one hour'
19 PRINT
20 PRINT
21 PRINT 'AUTHOR: Capt Damm J. Creagan'
22 PRINT 'Air Force Institute of Technology'
23 PRINT
24 PRINT 'OBJECTIVE: To teach the student now to use SUBROUTINES'
25 PRINT 'and LIBRARY functions.'
26 PRINT
27 PRINT
28 PRINT
29 PRINT
30 PRINT
31 INPUT 'Press the ENTER key to continue: '
32 END 21000
33 D SUB 30000
34 PRINT 'It's taking this part in its entirety.'
35 PRINT 'I wish to review selected areas.'
36 PRINT 'I want to go to the second part.'
37 PRINT 'I want to return to the Menu.'
38 PRINT
39 INPUT 'Press either capital A, B, C, or D and then press ENTER: '
40 IF T$ = 'A' GOTO 48000
41 IF T$ = 'B' GOTO 49000
42 IF T$ = 'C' GOTO 10000
43 IF T$ = 'D' GOTO 990
44 D SUB 299
45 D SUB 300
46 D SUB 301
47 D SUB 302
48 D SUB 21000
49 D SUB 30000
50 D SUB 300
51 D SUB 301
52 D SUB 302
53 D SUB 21000
54 D SUB 30000
55 D SUB 300
56 D SUB 301
57 D SUB 302
58 D SUB 21000
59 D SUB 30000
60 PRINT 'You are now done with this part of the lesson.  If you ENTER a '
61 PRINT 'ON', you will get on to the second part.  ENTER an 'O' for '
62 PRINT 'ON' to start over.  which do you want: A or B: '
63 INPUT ' THEN RUN
64 00
****** Listing of Program "LESSONS" ******

200 IF "A" = "A" THEN GOTO 980
230 GOTO 4800
240 INPUT "LESSONS 21:50"
250 INPUT "LESSONS 22:00"
260 PRINT 
270 PRINT "Please type in the number beside the area you wish" 
280 PRINT "to review (1 through 5) and then press ENTER - press 0 and" 
290 PRINT "press ENTER to return to the Menu."
300 PRINT 
310 INPUT "What is your choice?"
320 IF "A" = 1 GOTO 4800
330 PRINT "ON SCALE 2000,5000,4000,5000,8000"
340 GOTO 1000
350 GOTO 21000
360 PRINT "INTRODUCTION"
370 PRINT 
380 PRINT "In the last episode of our computerized book, we assigned you a" 
390 PRINT "homework problem that dealt with disk I/O disk input/output."
400 PRINT "It included most of the concepts that we have been studying."
410 PRINT "Now we are on the downhill part of the course. That's right."
420 PRINT "You are almost done with the hard parts of BASIC."
430 PRINT 
440 PRINT "The last obstacle is learning about SUBROUTINES, which you will"
450 PRINT "learn in the first half of this lesson. After that, we will" 
460 PRINT "review the numerous library functions (but not have to restring)"
470 PRINT "then, we'll just have to know how they work, and then go on"
480 PRINT "to lesson 6."
490 PRINT 
500 INPUT "press ENTER for"
510 INPUT "LESSON 21:50"
520 PRINT "INTRODUCTION"
530 PRINT 
540 PRINT "Somewhere during each lesson, we emphasize the value of doing"
550 PRINT "extra study besides what this course teaches you. We all learn:" 
560 PRINT "more when we DO something that we have read about. This is" 
570 PRINT "certainly true with learning a programming language."
580 PRINT 
590 PRINT "For now, though, get out your favorite BASIC manual, and curl" 
600 PRINT "up beside your computer for another lesson in BASIC."
610 PRINT 
620 INPUT "press ENTER for"
630 INPUT "LESSON 22:00"
640 IF "A" = 9 THEN GOTO 200
650 PRINT "SUBROUTINES"
660 PRINT

124
0020 PRINT
0030 PRINT "We use the word "ROUTINE" to describe the statements in the"
0040 PRINT "SUBROUTINE is used to describe"
0050 PRINT "a minisubprogram that was built, attached to the main program."
0060 PRINT "and used to perform a mini-task that, for some reason,"
0070 PRINT "needs special attention."
0080 PRINT
0090 PRINT "Usually, we use subroutines to do tasks which we perform"
0100 PRINT "often in our program such as printing out a menu after"
0110 PRINT "each module of a CAL program is finished by a student."
0120 PRINT "That way, we only have to write the main program once and call"
0130 PRINT "it when we need it."
0140 PRINT
0150 INPUT "Press ENTER."
0160 GOSUB 2000
0170 PRINT "Why would we want to use a subroutine?"
0180 PRINT
0190 PRINT "To do those parts of the program that are used often"
0200 PRINT "To go UNDER the normal routines."
0210 PRINT
0220 INPUT "Enter the letter opposite the correct answer: IT"
0230 PRINT
0240 IF E = "A" THEN GOTO 2155
0250 PRINT "WRONG - this was supposed to be an easy question to answer."
0260 PRINT "Obvioulsy, we screwed up somehow - we will send you back to the"
0270 PRINT "beginning of this part. Try reading between the lines a little."
0280 PRINT INPUT "Press ENTER."
0290 GOTO 2000

0300 PRINT "CORRECT - GOOD JOB!"
0310 PRINT
0320 INPUT "Press ENTER."
0330 GOSUB 2000

1455  PRINT "SUBROUTINES"
1460  PRINT
1470  PRINT "Subroutines are mini-programs that we put in our main programs."
1480  PRINT "and when we need them, we can go to them, perform the"
1490  PRINT "task they are built to do, and return to the main program."
1500  PRINT
1510  PRINT "Subroutines differ from other forms of program control in:
1520  PRINT "that they ALWAYS RETURN CONTROL TO THE STATEMENT THAT"
1530  PRINT "FOLLOWS THE STATEMENT THAT CALLED THEM. That means you"
1540  PRINT "can call a subroutine anywhere within a program, and then"
1550  PRINT "the computer will remember where the call came from, and when"
1560  PRINT "the subroutine has done its job, control will go back to"
1570  PRINT "the statement that followed the calling statement."
1580  PRINT
**** Listing of Program 'LESSONS' ****

1740 PRINT
1750 INPUT "press ENTER":T$ 
1760 GOSUB 21000
1770 PRINT "SUBROUTINES"
1780 PRINT
1790 PRINT "If you have built a program that has to continually print"
1800 PRINT "out a prompt, asking the users if they want to review"
1810 PRINT "previous sections of the program that have run, you would"
1820 PRINT "likely use a subroutine to ask the question, return to"
1830 PRINT "the main program with the answer stored in a variable,"
1840 PRINT "and branch to the right part of the program, based on the"
1850 PRINT "answer.
1860 PRINT
1870 PRINT
1880 PRINT "The following is an example such a program"
1890 PRINT
1900 INPUT "press ENTER":T$ 
1910 GOSUB 21000
1920 PRINT
1930 PRINT "We will explore the main points of this routine in the next "
1940 INPUT "section. Press ENTER to go on":T$ 
1950 PRINT
1960 GOSUB 21000
1970 PRINT "where does control transfer when a subroutine returns to the "
1980 PRINT "main program?"
1990 PRINT
2000 PRINT "The beginning of the program"
2010 PRINT "The calling statement"
2020 PRINT "The statement after the calling statement"
2030 PRINT "None of the above"
2040 PRINT
2050 INPUT "Enter the letter opposite the correct answer":T$ 
2060 PRINT
2070 IF T$ = "C" THEN GOTO 2680
2080 PRINT "WRONG - control is transferred to the statement after the "
2090 PRINT "calling statement. The correct answer is C"
2100 GOTO 2690
2110 PRINT "CORRECT - Super job ... Now we can go on"
2120 PRINT
2130 INPUT "press ENTER":T$ 
2140 GOSUB 40000
2150 IF T$ = "B" GOTO 3000
2160 RETURN
2170 GOSUB 21000
2180 PRINT "GOSUB 1 RETURN"
**Listing of Program 'LESSONS'**

```
400 PRINT
410 PRINT "The set of statements that you use to implement a subroutine"
420 PRINT "is made of the GOSUB and RETURN words. The GOSUB word is used"
430 PRINT "almost exactly like the GOTO statement. You put the line"
440 PRINT "number of the start of the subroutine on the right of the GOSUB"
450 PRINT "word. When the computer gets to it, it transfers control to"
460 PRINT "the subroutine. When the subroutine is done, it returns to the"
470 PRINT "main program by using the RETURN statement. You cannot GOTO"
480 PRINT "the main program from a subroutine without risking disaster."
490 PRINT "You should always use the RETURN statement."
500 PRINT
510 INPUT "Press ENTER"$*
520 GOSUB 21000
      " GOSUB 410"
530 PRINT
535 PRINT "Line 60 and line 90 call the subroutine, and line 10010 RETURNS"
540 INPUT "press ENTER"$* the appropriate statement. press ENTER$*
545 GOSUB 21000
550 GOSUB 4100
560 PRINT
570 PRINT "If line 21 calls the subroutine, what line gets control after"
580 INPUT "the RETURN statement? ENTER"$*
590 PRINT
600 IF T$ = "50" THEN GOTO 240
610 PRINT "The correct answer is line 50"
620 GOTO 240
630 PRINT "CORRECT - You are getting GOODIDOOGood!
640 PRINT
650 INPUT "Press ENTER"$*
660 GOSUB 21000
670 PRINT " GOSUB 1 RETURN"
680 PRINT
690 PRINT "You can have more than one GOSUB in a program, and you can"
700 PRINT "have more than one RETURN in a subroutine. If you have more"
710 PRINT "than one RETURN, then the computer will return when it reaches"
720 PRINT "the first RETURN statement it comes to. Generally speaking,"
730 PRINT "you should try to limit the number of exits from a subroutine"
740 PRINT "because it can get very confusing if you have RETURNS stuck"
750 PRINT "all over the place. It is usually possible to have only one"
760 PRINT "exit to an program or subroutine."
770 PRINT
780 INPUT "Press ENTER"$*
790 GOSUB 21000
800 PRINT "Is the following program valid?"
810 PRINT
820 PRINT "CHR$(14)"$ "Enter a number between 0 and 255"$*
830 PRINT
```

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

4450 PRINT"30 GSSUB 100"
4460 PRINT"40 END"
4470 PRINT"100 IF N < 10 THEN PRINT "CHR$(134)""Number is less than 10"CHR$(134)" ELSE GOTO 120"
4480 PRINT"110 RETURN"
4490 PRINT"120 IF N > 10 THEN PRINT "CHR$(134)""Number is more than 10"CHR$(134)" ELSE GOTO 140"
4500 PRINT"130 RETURN"
4510 PRINT"140 PRINT "CHR$(134)""Your number is 10"CHR$(134)"
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":IT$
4580 PRINT
4590 IF IT$ = "A" THEN GOTO 4610
4600 GOTO 4620
4610 PRINT"WRONG - the only problem is a confusing number of RETURNS"
4620 GOTO 4630
4630 PRINT"PRESS RETURN"
4640 GSSUB 21000
4650 PRINT"ENTER the command to go to a subroutine that starts on line"
4660 PRINT"100. Use line 100 and put one space between all terms.
4670 PRINT
4680 INPUT"ENTER the command now":IT$
4690 PRINT
4700 IF IT$ = "100 GSSUB 1000" THEN GOTO 4730
4710 PRINT"WRONG - the correct answer is : 100 GSSUB 1000"
4720 GOTO 4750
4730 PRINT"CORRECT"
4740 PRINT
4750 INPUT"PRESS ENTER":IT$
4760 GSSUB 40000
4770 IF IT$ = "B" THEN GOTO 4700
4780 RETURN
5000 GSSUB 11000
5010 PRINT
5020 PRINT
5030 PRINT"What do you think the output of the following is?"
5040 PRINT
5050 GSSUB 42000
5060 PRINT
5070 INPUT"PRESS ENTER for the answer":IT$
5080 GSSUB 21000
5090 PRINT"The answer is :"
5100 GSSUB 42000
***** Listing of Program 'LESSON5' *****
07/10/93 - 03:20:11

5110 PRINT
5120 PRINT"Main Program"
5130 PRINT"Subroutine One"
5140 PRINT"Subroutine Two"
5150 PRINT"Subroutine Two"
5160 INPUT" press ENTER";TS
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"310 RETURN"
5340 PRINT
5350 PRINT"Type out the line numbers as they would be executed, putting a" 
5360 PRINT"space between each number. (ie 10 20 30 40)"
5370 INPUT"Enter your answer";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 GOSUB 5420
5440 PRINT"CORRECT - gets complicated doesn't it?"
5450 PRINT
5460 INPUT" press ENTER";TS
5470 GOSUB 46000
5480 IF TS = "B" THEN GOTO 5000
5490 RETURN
5500 GOSUB 21000
5510 PRINT"ON GOSUB" 
5520 PRINT
5530 PRINT"Remember the ON GOTO statement from our previous lessons?"
5540 PRINT
5550 PRINT"There's a command very similar to ON GOTO that can be used with" 
5560 PRINT"subroutines. You have probably already guessed that the " 
5570 PRINT"command is called ON GOSUB."
Here is an example:

```
PRINT
PRINT:10 INPUT*CHR$(34):"ENTER A NUMBER BETWEEN 1 AND 4":N
PRINT:20 ON N GOSUB 300,400,500,600
PRINT
PRINT:140 IF N = 1 then subroutine 300 would be executed. If N = 2, subroutine 400 would execute, and so on
PRINT
150 PRINT
170 INPUT*PRESS ENTER*:T
180 GOSUB 21000
190 PRINT*ON GOSUB*
210 PRINT:10 INPUT*CHR$(34):"ENTER A NUMBER BETWEEN 1 AND 4":N
220 PRINT:20 ON N GOSUB 300,400,500,600
230 PRINT
240 PRINT:140 IF the value of N exceeds the number of options that are
250 PRINT"available, then ON GOSUB will default to the first available"
260 PRINT*line number :in this case it would be 300*
270 PRINT
280 PRINT
270 INPUT*PRESS ENTER FOR AN EXAMPLE*:T
290 GOSUB 21000
310 PRINT:10 PRINT*CHR$(34):"TYPE 1 FOR SQUARE TABLE":CHR$(34)
320 PRINT:20 PRINT*CHR$(34):"TYPE 2 FOR THE CUBE TABLE":CHR$(34)
330 PRINT:30 INPUT*A*
340 PRINT:40 ON A GOSUB 1000,2000
350 PRINT*50 GOTO 10*
370 PRINT:100% FOR X = 1 TO 50* 390 PRINT:500 RETURN*
370 PRINT:100% FOR X = 1 TO 50* 400 PRINT:1000 RETURN*
370 PRINT:100% FOR X = 1 TO 50* 410 PRINT:1500 PRINT X, X*X*X
370 PRINT:100% RETURN*
440 PRINT
450 PRINT*If you ENTER a 1, then subroutine 1000 is used. Which subroutine
460 INPUT*time is used if you ENTER a 4 :1000,2000 or NONE:"*:T
470 PRINT
480 IF T = "NONE" THEN GOTO 8510
490 PRINT"MORE - THE CORRECT ANSWER IS NONE"
500 GOTO 8520
510 PRINT"CORRECT"
520 PRINT
530 INPUT*PRESS ENTER*:T
540 GOSUB 42000
```
LISTING 0F PROGRAM "LESSONS" 05/01/83 - 03:20:11

3550 IF $S = "A" THEN GOTO 3600
3560 RETURN
3680 REM **
3690 REM ** This subroutine clears the screen on any terminal
3695 REM **
3700 FOR X = 1 TO 24
3705 PRINT
3710 NEXT X
3720 RETURN
3725 REM **
3730 REM ** This subroutine is the menu
3735 REM **
3760 PRINT " LESSON 5"
3765 PRINT
3767 PRINT "This is the first part of a two part lesson"
3768 PRINT "It is divided into the following sections:"
3770 PRINT
3775 PRINT "1) Introduction"
3780 PRINT "2) Nested Subroutines"
3785 PRINT "3) Subroutines"
3790 PRINT "4) ON GOSUB"
3795 PRINT "5) GOSUB & RETURN"
3800 PRINT
3805 PRINT
3810 PRINT
3815 PRINT
3820 REM **
3821 REM ** This subroutine gives the student a chance to review the lesson
3826 REM **
3836 REM **
3840 GOSUB 21000
3845 PRINT "Which do you wish to do?"
3850 PRINT
3855 PRINT "A Continue on"
3860 PRINT "B Review this section again"
3865 PRINT
3870 INPUT "$S" press the letter opposite the correct answer and press ENTER:"
3875 IF $S = "A" OR $S = "B" THEN RETURN
3880 GOSUB 42000
3885 PRINT
3890 REM **
3891 REM ** This subroutine is for the gosub example
3896 REM **
3906 REM **
3916 PRINT "A CAl program is a computer assisted instruction"
3920 PRINT "program that teaches students."
3925 PRINT "Look use sub. after every major part:
3930 PRINT "Look use sub. after every major part:"
3935 PRINT "End You must ensure program doesn't goto sub by accident"
3940 PRINT
***** Listing of Program 'LESSON5' *****

41060 PRINT"110 REM the subroutine follows"
41063 PRINT"120 REM"
41065 PRINT"10000 INPUT"CHR$(34)//Do you want to continue or review (A=Com B=Rev)"CHR$(34)//;176"
41070 PRINT"10010 RETURN"
41080 RETURN
42000 PRINT"10 PRINT"CHR$(34)//Main Program"CHR$(34)
42010 PRINT"20 GOSUB 100"
42020 PRINT"30 GOSUB 200"
42030 PRINT"40 END"
42040 PRINT"100 PRINT"CHR$(34)//Subroutine One"CHR$(34)
42045 PRINT"110 GOSUB 200"
42050 PRINT"120 RETURN"
42060 PRINT"200 PRINT"CHR$(34)//Subroutine Two"CHR$(34)
42070 PRINT"210 RETURN"
42080 RETURN
48000 RUN *MENU*
49000 RUN"lesson5a"
50000 END
**Listing of Program “LESSON5A”**

```
100 GOSUB 21000
200 GOSUB 31000
290 PRINT "I'm taking this part in its entirety.
300 PRINT "I wish to review selected areas (or take the test).
310 PRINT "I want to go to the first part."
320 PRINT "I want to return to the Menu."
330 PRINT
340 INPUT "Press either capital A, B, C, or D and then press ENTER: IT#
350 IF T# = "B" GOTO 48000
360 IF T# = "C" GOTO 49000
370 IF T# = "D" GOTO 1900
380 IF T# = "A" GOTO SPC
450 GOSUB 21000
550 GOSUB 3000
600 GOSUB 4000
750 GOSUB 5000
770 GOSUB 7000
773 GOSUB 9200
980 GOSUB 21000
470 GOTO 49000
500 GOSUB 11000
102 GOSUB 20000
150 PRINT
160 PRINT "Please type in the number beside the area you wish."
170 PRINT "to review (1 through 5) and then press ENTER - press J and"
180 PRINT "press ENTER to return to the Menu."
190 PRINT
200 INPUT "What is your choice? IN"
210 IF Y = "A" GOTO 48000
220 IF Y = "B" GOTO 49000
230 IF Y = "C" GOTO 1900
240 IF Y = "D" GOTO 21000
250 PRINT "INTRODUCTION"

260 PRINT "In this lesson we will tackle the different categories and"
270 PRINT "kinds of functions that are available to us in Microsoft BASIC.
280 PRINT "We will use principles already covered in previous"
290 PRINT "sessions. If you have trouble with some of the concepts, then"
300 PRINT "you will have to review the appropriate lesson to catch up."
310 PRINT
320 PRINT "However, we will not be going into great depth with our"
330 PRINT "explanations. Many functions have very specific uses and it"
340 PRINT "would not be useful for you to memorize them. The idea you"
350 PRINT "should get from this part is that there are many functions"
360 PRINT "available, and that when you need them, you should get out your"
370 PRINT "manual and look to the specific implementation of each one."
```
.... Listing of Program 'LESSON5A' ....

2160 PRINT
2170 INPUT"press ENTER":T$
2180 RETURN
2200 GOSUB 21000
2310 PRINT* Functions Overview*
2320 PRINT
2330 PRINT"Remember in the second part of Lesson 1 when we described"
2340 PRINT"FUNCTIONS for the first time? We said that we would come back"
2350 PRINT"to them in another lesson. Well, this is it!"
2360 PRINT
2370 PRINT"From that lesson we should remember that functions are pre-
2380 PRINT"written instructions that perform commonly used operations."
2390 PRINT"You can look at functions like they were mini-subroutines."
2400 PRINT"only you don't use GOSUB or RETURN statements to call them"*
2410 PRINT"Instead, you just use the keyword associated with the FUNCTION"
2420 PRINT"and the computer performs the appropriate operation auto-
2430 PRINT"matically. In the next sections we will study two types of"
2440 PRINT"functions, Library, and User Functions."
2450 PRINT
2460 INPUT"press ENTER":T$
2470 GOSUB 21000
2480 PRINT* Functions Overview*
2490 PRINT
2500 PRINT"Library functions contain useful operations that have been"
2510 PRINT"written and stored in the computer, and are there whenever you"
2520 PRINT"need them. User functions are functions that you make up"
2530 PRINT"by inserting the instruction to make them in your program."
2540 PRINT"Then, when your program needs the special USER FUNCTION, it"
2550 PRINT"can call on it with a special word."
2560 PRINT
2570 PRINT"There are many functions, and depending on the specific"
2580 PRINT"implementation of Microsoft BASIC, you probably have at least"
2590 PRINT"12 Library functions stored in your computer. On the next"
2600 PRINT"screen is a list of the typical set of Library functions."
2610 PRINT
2620 INPUT"press ENTER":T$
2630 GOSUB 21000
2640 PRINT* Typical Library Functions*
2650 PRINT
2660 GOSUB 41300
2670 PRINT
2680 PRINT"Specific examples of these functions will be given later, or"
2690 PRINT"you may look them up in your BASIC manual."
2700 INPUT"press ENTER":T$
2710 GOSUB 21000
2720 PRINT"Is the following statement TRUE or FALSE?"
**Listing of Program 'LESSON5A'**

2430 PRINT
2440 PRINT "Library functions are predefined instructions that are stored"" inside the computer, and may be used at any time in your program.
2450 PRINT
2470 PRINT
2480 PRINT 'A TRUE'
2490 PRINT 'B FALSE'
2500 PRINT
2510 INPUT ENTER the letter opposite the correct answer: IT
2520 PRINT
2530 IF T$ = "A" THEN GOTO 2560
2540 PRINT 'WRONG - the sentence is good. correct answer is A'
2550 GOTO 2570
2560 PRINT 'CORRECT'
2570 PRINT
2580 INPUT press ENTER: IT
2590 GOSUB 21000
2600 PRINT 'Is the following sentence TRUE or FALSE'
2610 PRINT
2620 PRINT 'User Functions are made by the user and are not normally stored'
2630 PRINT 'in the computer.'
2640 PRINT
2650 PRINT 'A TRUE'
2660 PRINT 'B FALSE'
2670 PRINT
2680 INPUT ENTER the letter opposite the correct answer: IT
2690 PRINT
2700 IF T$ = "A" THEN GOTO 2730
2710 PRINT 'WRONG - the statement is true. the correct answer is A'
2720 GOTO 2740
2730 PRINT 'CORRECT'
2740 PRINT
2750 PRINT press ENTER: IT
2760 GOSUB 40000
2770 IF T$ = "B" THEN GOTO 2790
2780 RETURN
2790 GOSUB 21000
3010 PRINT 'Library Functions'
3020 GOSUB 41000
3030 GOSUB 41000
3040 PRINT
4050 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.'
4060 PRINT
4120 PRINT press ENTER: IT
4120 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 42900
4230 PRINT
4240 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**:TS
4310 GOSUB 21000
4320 PRINT* Library Functions
4330 PRINT
4340 PRINT
4350 PRINT*Notice that the function allows you to use a variable as an
4360 PRINT*argument for the number you want the square root of. In *
4370 PRINT*addition to this, you can use a FUNCTION as an argument for a *
4390 PRINT*FUNCTION**
4400 PRINT
4410 INPUT*Press ENTER* for an example**:TS
4420 GOSUB 21000:
4430 PRINT* Library Functions
4440 PRINT
4450 GOSUB 43600
4460 PRINT
4470 PRINT*In this example, no matter what the sign of the number you *
4480 PRINT*enter, you will get a valid square root, because ABS(N) will*
4490 PRINT*give the absolute value of N :N as a positive number, and then*
4500 PRINT*SOR(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.*
4540 PRINT
4550 PRINT
4560 INPUT*Press ENTER**:TS
4570 GOSUB 21000:
4580 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' 07/10/83 - 02:48:30

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 GOSUB 21000
4560 PRINT"Library Functions"
4561 PRINT  
4562 PRINT"The expression ABS(X) is evaluated first, then the outside"
4563 PRINT"expression - SQRT(---) - is evaluated next. Remember from the"
4564 PRINT"first lesson when we said that the parenthesis is the highest"
4565 PRINT"priority arithmetic expression? That means that any expression"
4566 PRINT"that is within parenthesis will be evaluated first. If more"
4567 PRINT"than one set of parenthesis is used, then the expression with-"
4568 PRINT"in the inner-most set of parenthesis is evaluated first."
4569 PRINT
4570 PRINT
4571 PRINT"Is the following sentence TRUE or FALSE?"
4572 PRINT"You can write your own library functions in special cases."
4573 PRINT
4574 PRINT"A TRUE"
4575 PRINT"B FALSE"
4576 PRINT
4577 INPUT"ENTER the letter opposite the correct answer": T$  
4578 PRINT
4579 IF T$ = "A" THEN GOTO 4830
4580 PRINT"WRONG - library functions are permanently stored in the "
4581 PRINT"computer and cannot be created, the correct answer is B"
4582 GOTO 4840
4583 PRINT"CORRECT"
4584 PRINT
4585 INPUT"press ENTER": T$
4586 GOSUB 21000
4587 PRINT"In the next few screens, we will ask you questions concerning"
4588 PRINT"library functions. You should get out your BASIC manual and"
4589 PRINT"look up the answer to the questions before you attempt to enter"
4900 PRINT "your response."
4910 PRINT
4920 PRINT "You should give your answer in the format, FUNCTION(number)."
4930 PRINT "You will always be given the number for the function, and you"
4940 PRINT "do not have to include a line number. For example, if we ask"
4950 PRINT "for the function that gives the square root of 10, you would"
4960 PRINT "use in: SQR(10)."
4970 PRINT
4980 INPUT "press ENTER":T$  
4990 GOSUB 21000
5000 INPUT "What is the function that gives the natural log of 9":T$
5010 PRINT
5020 IF T$ = "LOG(9)" THEN PRINT "CORRECT" ELSE PRINT "WRONG - ANSWER is LOG(9)"
5030 PRINT
5040 INPUT "Press ENTER":T$
5050 GOSUB 21000
5060 INPUT "Name the function that gives the absolute value of -3":T$
5070 PRINT
5080 IF T$ = "ABS(-3)" THEN PRINT "CORRECT" ELSE PRINT "WRONG - the answer is ABS(-3)"
5090 PRINT
5100 INPUT "Press ENTER":T$
5110 GOSUB 21000
5120 INPUT "What is the function that gives the arctangent of 4":T$
5130 PRINT
5140 IF T$ = "ATN(4)" THEN PRINT "CORRECT - GREAT JOB" ELSE PRINT "WRONG - the correct answer is ATN(4)"
5150 PRINT
5160 INPUT "Press ENTER":T$
5170 GOSUB 21000
5180 INPUT "What is the function that gives the sine of .5":T$
5190 PRINT
5200 IF T$ = "SIN(.5)" THEN PRINT "CORRECT" ELSE PRINT "WRONG - the correct answer is SIN(.5)"
5210 PRINT
5220 INPUT "Press ENTER":T$
5230 GOSUB 21000
5240 INPUT "Name the function to give a random number between 0 & 1":T$
5250 PRINT
5260 IF T$ = "RND(0)" THEN PRINT "CORRECT - good one" ELSE PRINT "WRONG - the correct answer is RND(0)"
5270 PRINT
5280 INPUT "Press ENTER":T$
5290 GOSUB 40000
5300 IF T$ = "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"
Listing of Program 'LESSMONA'

5040 PRINT "when you can't find a library function that will do the job."
5050 PRINT
5060 PRINT "The statement you use to DEFINE a user function, so the"
5070 PRINT "computer will know what you are doing, is called the DEF state-
5080 PRINT "ment. The user function only applies to the program in which"
5090 PRINT "it was defined. When the program is abandoned, the function is"
5100 PRINT "no longer valid. An example of a user defined function is"
5110 PRINT "coming up .... but first, something a little different."
5120 PRINT
5130 PRINT
5140 INPUT 'press ENTER'; T$
5150 GOSUB 21000
5160 PRINT "Can user functions be carried over from one program to another?"
5170 PRINT
5180 PRINT 'A  Yes'
5190 PRINT 'B  No, they always have to be redefined'
5200 PRINT 'C  Both A & B above.'
5210 PRINT
5220 INPUT "Enter the letter opposite the correct answer";' T$
5230 PRINT
5240 IF T$ = 'B' THEN GOTO 155
5250 IF T$ = 'C' THEN GOSUB 21000: PRINT "Are you related to my wife";' PRINT "Please take another choice....."
5260 IF T$ = 'C' THEN GOSUB 2100: PRINT "Wrong - the correct answer is B"
5270 GOTO 115
5280 PRINT 'Correct'
5290 PRINT
5300 GOSUB 21000
5310 PRINT "There are two user functions in this program. They are define-
5320 PRINT "d in lines 10 and 20, and called on in lines 30, and 50. We will"
5330 PRINT "cover this program in greater detail in the next section."
5340 PRINT
5350 INPUT 'press ENTER'; T$
5360 GOSUB 44000
5370 PRINT "DEF Statement"
5380 PRINT "The DEF statement is used to define a user function and it has"
5390 PRINT "the following format:"
7060 PRINT"Line 10: DEF FN(func. name)(vars) = (exp)"
7070 PRINT
7080 PRINT"You must never use the DEF statement in the IMMEDIATE mode."
7090 PRINT"You always have to have a line number. In CPM and Cromemco"
7100 PRINT"systems, you must separate the terms DEF and FN by one space."
7110 PRINT"In TRS-80s you don't have to. The (func. name) is any valid"
7120 PRINT"variable name. (vars) is an optional parameter that may be"
7130 PRINT"passed to the function. If it is used in the DEF statement, it"
7140 PRINT"MUST be used when it is called in. (more on that later). The"
7150 PRINT"term (exp) is the calculation that you wish the function to do."
7160 PRINT
7170 INPUT"Press ENTER":T$ 
7180 GOSUB 21000 
7190 GOSUB 44000 
7200 PRINT
7210 PRINT"Set out your BASIC manual and look up DEF. It will show you"
7220 PRINT"examples similar to this. Line 10 defines a function named"
7230 PRINT"RC that will be set equal to the expression on the right side"
7240 PRINT"or the statement. - it will return a random number between 1"
7250 PRINT"and 10. Note that this user function uses a library function"
7260 PRINT"as part of its definition. This is legal."
7270 PRINT
7280 INPUT"Press ENTER":T$ 
7290 GOSUB 21000 
7300 GOSUB 44000 
7310 PRINT
7320 PRINT"Can you use a library function as part of the definition of a "
7330 PRINT"USER function?"
7340 PRINT
7350 INPUT"A YES"*
7360 INPUT"B NO" *
7370 PRINT
7380 INPUT"ENTER the letter opposite the correct answer":T$ 
7390 PRINT
7400 IF T$ = "A" THEN GOTO 7315.
7410 PRINT"WRONG - you can use a library function inside a USER function"
7420 GOTO 7317 
7430 PRINT"CORRECT"
7440 PRINT
7450 INPUT"Press ENTER":T$ 
7460 GOSUB 21000 
7470 GOSUB 44000 
7480 PRINT
7490 PRINT"Line 70 calls on the function defined in line 10. When it"
7500 PRINT"does, it is set equal to a random number between 1 and 10 and"
7510 PRINT"then it is printed out in line 40. Line 50 calls on the "
7520 PRINT"function in line 20, but it sends two values to the DEF state-"
7530 PRINT"ment. It sends 1 (a random number) and the number 4. Ah!"
"A valid variable can be passed to a function, even a string."
740 PRINT
741 INPUT "Press ENTER":T$  
742 GOSUB 21000  
743 GOSUB 44000  
744 PRINT
745 PRINT "A string could be passed in a function, but it would have to be operated on legally within the DEF statement. In this case"
746 PRINT "the DEF statement in line 20 expects two numerical variables."
747 PRINT "Note that the two variables passed in line 50 do not match"
748 PRINT "the defined variables. This is also legal. You can view the"
749 PRINT "DEF statement as its own little program, it does not know the"
750 PRINT "value of any of the variables in the outside program."
751 INPUT "Press ENTER":T$  
752 GOSUB 21000  
753 GOSUB 44000  
754 PRINT
755 PRINT "The two values that are defined as A. and B will be set equal"
756 PRINT "to the corresponding values of X and 4. A will equal X and B"
757 PRINT "will equal 4. The DEF statement will then use these numbers to"
758 PRINT "calculate the (x^2) part of the statement, and then Y will be"
759 PRINT "made equal to this value. Finally, the value will be printed"
760 PRINT "in line 50. You may wish to copy a program like this and"
761 PRINT "exeriment with it to see how it works."
762 INPUT "Press ENTER":T$  
763 GOSUB 21000  
764 GOSUB 44000  
765 PRINT
766 PRINT "What value would be passed to B in the second DEF statement?"
767 PRINT
768 PRINT "The value of X"
769 PRINT "Cannot tell - not enough information"
770 PRINT
771 INPUT "Enter the letter opposite the correct answer":T$  
772 PRINT
773 IF T$ = "A" THEN GOTO 774a  
774a PRINT "Wrong - the correct answer is A"  
775 GOTO 774a  
776 PRINT "Correct"
777 PRINT
778 INPUT "Press ENTER":T$  
779 GOSUB 21000  
780 GOSUB 44000  
781 PRINT
782 PRINT "Here is an example of passing a string in a user function."
In all the examples that we have shown, we have always told
the computer when we wanted to invoke a user function by using
the prefix "FN: vari". That is the only way to get to your
user function.

Press enter to submit your answer when you
involve a user function.

INPUT press enter*78
GOSUB 21000
GOSUB 45000
PRINT

Note in the example above, that we passed a string within a
numeric variable, but that in the DEF statement, we set the
numeric variable equal to LEN(string). LEN(string) is an
example of yet another type of FUNCTION that we will study in
the next lesson. LEN(string) returns a numeric value equal to
the LENGTH of the string that is within the parenthesis.

Don't worry about it now, we will cover it in the next lesson.
However, because it is a NUMERIC value, its type matches the
PRINT statement, and therefore it is valid.

PRINT
INPUT press enter*78
GOSUB 21000
PRINT "What kind of function do you create with the DEF statement?
PRINT "A USER"
PRINT "A LIBRARY"
PRINT "EITHER USER or LIBRARY"
PRINT "DEFINED FUNCTIONS"
PRINT
INPUT press enter the letter opposite the correct answer*78
IF T$ = "A" THEN 7840
PRINT "Wrong - only USER defined functions are created with the DEF"
PRINT "statement. The correct answer is A"
GOTO 7850
PRINT "CORRECT! - but that was an easy one. try the next question :"
PRINT "you think you are up to it?"
PRINT "we know you are - we're just trying"
PRINT "to lighten things up a bit:"
PRINT
INPUT press enter*78
GOSUB 21000
PRINT "the following statement valid"
PRINT
PRINT "DEF FN89(\(A,B\)) = A + B"
PRINT
PRINT "A Yes, to call on it simply but FN88 somewhere in your program"
### Listing of Program 'LESSON5A' ####

7965 PRINT"No, the terms A and B are not defined and will cause errors"
7969 PRINT"No, the syntax is good, but the beat is poor, you can't "
7970 PRINT"dance to it."
7979 PRINT
7980 INPUT"Enter the letter opposite the correct answer";T$
7981 PRINT
7982 IF T$ = "E" THEN PRINT"O.K wise guy, even my 8 year old knew that was":PRINT"a stupid answer. Try again.":PRINT:INPUT"press ENTER";T$:GOSUB 21000:GOTO 7980
7983 IF T$ = "C" THEN GOTO 7996
7984 PRINT"WRONG - the correct answer is C"
7985 GOTO 7997
7986 PRINT"CORRECT"
7987 PRINT
7988 INPUT"press ENTER";T$:GOSUB 40000
7989 GOSUB 40000
7990 IF T$ = "B" THEN GOTO 7000
7991 RETURN
7992 GOSUB 21000
7993 PRINT"You have completed this lesson and now you can go to the test"
7994 PRINT"If you wish to review parts of this lesson, type in an 'R'."
7995 PRINT"If you wish to continue to the test, type in a 'C'."
7996 PRINT
7997 INPUT"Enter your choice : R or C";T$:GOSUB 40000
7998 PRINT
7999 IF T$ = "R" THEN RUN
8000 IF T$ = "C" THEN GOTO 47000
8001 GOTO 9020
8002 REM **
8003 REM ** This subroutine clears the screen on any terminal
8004 PRINT 1000:FOR X = 1 TO 24
8005 NEXT X
8006 RETURN
8007 PRINT"This is the second part of a two part lesson"
8008 PRINT"It is divided into the following sections."
8009 PRINT
8010 PRINT"1: Introduction"
8011 PRINT"2: User Defined Functions"
8012 PRINT"3: Functions Overview"
8013 PRINT"4: DEF Statements"
8014 PRINT"5: Library Functions"
8015 PRINT"6: TEST"
8016 PRINT
8017 RETURN
8018 RETURN
8019 RETURN
8020 RETURN
8021 RETURN
8022 RETURN
8023 PRINT
**Listing of Program 'LESSON5A'**

```
4000.1 GOSUB C1000
40010 PRINT"Which do you wish to do?"
40011 PRINT
40012 PRINT"A. Continue on"
40013 PRINT
40014 PRINT"B. Review this section again"
40015 PRINT
40016 PRINT"C. Press the letter opposite the correct answer and press ENTER":TS
40017 IF TS = "A" OR TS = "B" THEN RETURN
40018 GOTO 300000
40020 REM **
40021 REM ** these are for library functions examples
40022 REM **
40023 IF A10 PRINT"ABS(exp) - gives absolute value. 7) LOG10(exp) - gives LOG10 of"
40024 IF A10 PRINT"ATN(exp) - arctangent in radians 8) RAND01 - random numbers"
40025 IF A10 PRINT"COS(exp) - returns cosine of exp 9) SGN(exp) - sign of exp"
40026 IF A10 PRINT"EXP(exp) - natural exponential 10) SIN(exp) - sine of exp"
40027 IF A10 PRINT"FUP(exp) - gives integer of exp 11) SQR(exp) - square root"
40028 IF A10 PRINT"INT(exp) - gives integer of exp 12) TAN(exp) - tangent of exp"
40029 IF A10 PRINT
40030 PRINT"= any appropriate numeric expression - remember, you"
40031 IF A10 PRINT"can't use ANY number for some of the functions (SQR(-1) does'nt)
40032 IF A10 RETURN
40033 REM **
40034 REM ** the subroutine is for the library function area
40035 REM **
40036 IF A10 INPUT"CHR$(24)";T in the number you want the square root of"CHR$(24):IN"
40037 IF A10 PRINT"STOR"IN"
40038 IF A10 PRINT"GOTO 10"
40039 IF A10 RETURN
40040 REM **
40041 REM ** this is another example for the library functions
40042 REM **
40043 IF A10 INPUT"CHR$(24)";Enter the number you want the square root of"CHR$(24):IN"
40044 IF A10 PRINT"STOR"IN"
40045 IF A10 PRINT"GOTO 10"
40046 IF A10 RETURN
40047 REM **
40048 REM ** this example is for user functions
40049 REM **
40050 PRINT"10 DEF FHR2 = INT(RND01) * 10"
40051 PRINT"10 PRINT"DEF FWR = A + B/2 + (A - B)"
40052 PRINT"10 PRINT"X = FHR2"
40053 PRINT"10 PRINT X" Y = FWR
40054 PRINT"10 PRINT Y" X = FWR
40055 PRINT"10 PRINT X"
40056 PRINT"10 PRINT "END"
```

214
**** Listing of Program 'LESSONS' ****

44060 RETURN
44970 REM **
44980 REM ** example of DEF with strings
44990 REM **
45000 PRINT"10 DEF FNX(AS) = LEN(AS)"
45010 PRINT"20 AS = "CHR$(I);"This is a function"CHR$(J)
45020 PRINT"30 X = FNX(AS)"
45030 PRINT"40 PRINT X"
45040 RETURN
47000 GOSUB 21000
47110 PRINT"Going to test number 5 - wait patiently please"
47320 RUN "TESTS"
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 CREGAN
1030 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM ** N$(X) = NAMES ARRAY. USED TO READ IN SEQUENTIAL NAMES. AND TO WRITE OUT
1070 REM ** UPDATE NAMES.
1080 REM ** S(x) = SCORES ARRAY - USED TO READ AND WRITE SCORES
1090 REM ** Q(x) = ARRAY TO KEEP TRACK OF NUMBER OF ELEMENTS EQUALS 1, THE ANSWER WAS CORRECT
1100 REM **
1110 REM ** CLEAR 3000
1120 DIM N$(1000)
1130 DIM Q(100)
1140 DIM S(1:1000)
1150 PRINT

FINAL TEST :lesson 3:

1220 PRINT

1230 PRINT "This test consists of 10 questions, you must get 70 percent."
1240 PRINT "of these correct to pass. (that’s 7 right out of the 10 questions.
1250 PRINT "Solutions: Use only capital letters in your answers, don’t"
1260 PRINT "include extra spaces or letters. If you successfully complete"
1270 PRINT "the test, you can go on to the last lesson."
1280 PRINT "GOOD LUCK"
1290 PRINT
1300 INPUT "press ENTER to continue":IT$
1310 GOSUB 4120
1320 PRINT "The order in which the following lines will be executed:"
1330 PRINT "leave one space between each line number - i.e., if the execution"
1340 PRINT "sequence is ten, twenty and thirty, then type in 10 20 30"
1350 PRINT
1360 PRINT "10 GOSUB 1000"
1370 PRINT "20 PRINT "CHR\&44\&"DONE"\&CHR\&44."
1380 PRINT "30 END"
1390 PRINT "1000 RETURN"
1400 PRINT
1410 INPUT "Enter the sequence now":IT$
1420 PRINT
1430 IF IT$ = "10 1000 20 30" THEN GOTO 1470
1440 PRINT "Wrong - the correct answer is 10 1000 20 30"
1450 PRINT "See part I. Subroutines"
**** Listing of Program 'TESTS' ****

1460 GOTO 1470
1470 PRINT "CORRECT"
1480 Q11 = 1
1490 PRINT
1500 INPUT "Press ENTER to exit"
1510 GOSUB 4120
1520 PRINT "What will happen when you input then number 4 to the following:
1530 PRINT "program?"
1540 PRINT
1550 PRINT "0 INPUT A"
1560 PRINT "DO ON A GOSUB 1000,2000,3000"
1570 PRINT "DO END"
1580 PRINT ".... rest of 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 T# = "S" THEN GOTO 1700
1680 PRINT "WRONG - the correct answer is D"
1690 PRINT "I If you don't satisfy one of the 3N GOSUB choices"
1700 PRINT "the line defaults to the line just below it."
1710 PRINT "See part 1. On GOSUB"
1720 GOTO 1730
1730 PRINT "CORRECT"
1740 Q11 = 1
1750 PRINT
1760 INPUT "Press ENTER to exit"
1770 GOSUB 4120
1780 PRINT "How many RETURNS can you have in a subroutine?"
1790 PRINT
1800 PRINT "A One"
1810 PRINT "C As many as you want, but they should be used to a minimum"
1820 PRINT "D One (or even zero) GOSUB"
1830 PRINT "G No more than the amount of space available"
1840 PRINT
1850 GOSUB 4120
1860 PRINT
1870 IF T# = "C" THEN GOTO 1930
1880 PRINT "WRONG - the correct answer is G"
1890 PRINT "See part 1. Subroutines"
1900 GOTO 1910
1910 PRINT "CORRECT"
#### Listing 2 - Program "TESTS" ####

```plaintext
1720 3.0: = 1
1730 PRINT
1740 INPUT"press ENTER"+1
1750 GOSUB 4100
1760 PRINT"What is wrong with this program?"
1770 PRINT
1780 PRINT"Do INPUT A"
1790 PRINT"Do GOSUB 1000;"
2000 PRINT"END";
2010 PRINT"1000 IF A = 10 THEN RETURN"
2020 PRINT"1000 IF A = 11 THEN RETURN"
2030 PRINT"1010 GOTO 10";
2140 PRINT
2150 PRINT"4 Nothing"
2160 PRINT"5 One of the possible exits from the subroutine is incorrect"
2170 PRINT"6 There are too many RETURN statements"
2180 PRINT"D The IF statements aren't allowed in a subroutine like this;";
2190 PRINT
2200 PRINT
2210 IF A = 9 THEN GOTO 2190
2220 PRINT"ACK! The correct answer is 9"
2240 PRINT"If A is greater than 10, then line 1111 causes;"
2250 PRINT"the subroutine to branch into the main program;";
2260 PRINT"That will eventually cause the computer to get;"
2270 PRINT"A BS and ROM. See part 2. Subroutines;"
2280 GOTO 110
2290 PRINT"Correct;"
2300 J = 1
2310 GOSUB
2320 INPUT"press ENTER"+
2330 GOSUB 4100
2340 PRINT"For the next few questions, you should be sure you have your;"
2350 PRINT"BASIC manual available so you can figure them out correctly"
2360 PRINT"All the questions are from part 1 of the lesson, and your;"
2370 PRINT"BASIC manual;"
2380 PRINT
2390 PRINT
2400 INPUT"press ENTER"+
2410 GOSUB 4100
2420 PRINT"What is the value of the following statement;"
2430 PRINT
2440 PRINT"206 D5;"
2450 PRINT
2460 PRINT"F;"
2470 PRINT"G;"
2480 PRINT"D the natural logarithm of 10;";
```

---

219
**** Listing of Program 'TESTS' ****

CLS
PRINT "100"
200 PRINT
300 sleep 400
400 PRINT
420 IF T# = 'A' THEN GOTO 440
420 PRINT"WONG - the correct answer is -"
440 GOTO 420
450 PRINT"CORRECT"
460 C# = 1
470 PRINT
490 INPUT press ENTER
500 SCENE 3111
510 PRINT"What type of functions would be written by you?"
500 PRINT"Type in your answer using all capital letters. Do NOT add "
500 PRINT"the word FUNCTION on the end of your answer."
500 PRINT
500 INPUT"Use in your answer now:"
500 PRINT
570 IF T# = 'USER' THEN GOTO 260
580 PRINT"WONG - the correct answer is USER"
590 GOTO 260
600 PRINT"CORRECT"
610 C# = 1
620 PRINT
630 INPUT press ENTER
640 SCENE 4111
650 PRINT"What is the term you would use to define a USER function?"
660 PRINT"Hint: It's two words. DON'T include a variable. Leave a space."
670 PRINT"between the two words:"
680 PRINT
690 INPUT"Enter your answer now:"
700 PRINT
710 IF T# = 'DEF FN' THEN GOTO 270
720 PRINT"WONG - the correct answer is DEF FN"
730 GOTO 270
740 PRINT"CORRECT"
750 C# = 1
760 PRINT
770 INPUT press ENTER
780 SCENE 5111
790 PRINT"What is the term that could complete line 2 if we wished to"
800 PRINT"call in the USER function in line 10?"
**** Listing of Program "TEST" ****

100 PRINT
110 INPUT "ENTER your answer now:"
120 PRINT
130 IF S = "FINISH" THEN GOTO 200
140 PRINT "Wrong - the correct answer would be FINISH"
150 GOTO 100
160 PRINT "CORRECT"
170 M = 1
180 PRINT
190 INPUT "Press ENTER to exit"
200 SUB A(1)
210 PRINT "Which of the following functions is valid?"
220 PRINT
230 PRINT "A) DEF FNAME(L,M) = Y + Z + "
240 PRINT "B) DEF FNAME(L,M) = Y + Z - "
250 PRINT "C) DEF FNAME(L,M) = Y + Z + (^"
260 PRINT "D) DEF FNAME(L,M) = Y + Z - (^"
270 PRINT "E) DEF FNAME(L,M) = Y + Z + (^"
280 PRINT "F) DEF FNAME(L,M) = Y + Z - (^"
290 PRINT "G) DEF FNAME(L,M) = Y + Z + (^"
300 PRINT
310 PRINT "A) "S"
320 PRINT
330 IF S = "A" THEN GOTO 200
340 PRINT "Wrong - the correct answer is B"
350 GOTO 100
360 PRINT "CORRECT"
370 C = 1
380 PRINT
390 INPUT "Press ENTER to exit"
400 SUB A(1)
410 PRINT "What is the proper statement to exit from a subroutine?"
420 PRINT
430 IF S = "FINISH" THEN GOTO 200
440 PRINT "Wrong - the correct answer is FINISH"
450 PRINT "See part 1. Subroutines"
460 GOTO 100
470 PRINT "CORRECT"
480 C = 1
490 PRINT
500 INPUT "Press ENTER to exit"
510 SUB A(1)
520 FOR I = 1 TO 10
530 Y = Y + 0.1
540 NEXT I
550 PRINT "You have finished the test, out of 10 possible correct answers!"
**** Listing of Program "TESTS" ****

10 IF X >= 6 THEN PRINT "YOU HAVE PASSED"
20 GOTO 3000
30 IF X >= 6 THEN GOTO 3420
35 PRINT "YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
40 PRINT
45 PRINT "YOU SHOULD RETAKE LESSON 5"
50 PRINT
55 PRINT "YOU will be returned to the menu."
60 PRINT
65 GOTO 4200
70 PRINT
75 PRINT "Do you want your score recorded on a permanent file?"
80 PRINT
85 PRINT "YES"
90 PRINT "NO"
95 PRINT
100 INPUT "Which file?"
110 IF Y = "S" THEN GOTO 3070
120 GOSUB 4110
130 PRINT "To record your score, we must open a file and put your name"
140 PRINT "in it. Therefore, surprisingly, we need your name. If your"
150 PRINT "name is not unique among the students likely to take this test,"
160 PRINT "please contact your test monitor for an identifying word that"
170 PRINT "will make you unique. Then enter that word below."
180 PRINT
190 PRINT "If you have already entered a score previously, be sure to"
200 PRINT "enter the same name you used before. Please use all capitals."
210 PRINT
220 INPUT "ENTER your word or name now!"
230 OPEN 11, "SCORES"
240 I = 
250 IF Z = "S" THEN GOTO 5600
260 L = 
270 INPUT "NAME:",
280 INPUT "WORD:",
290 IF W = "S" THEN GOTO 3570
300 GOTO 5600
310 CLOSE
320 X = 
330 PRINT "GO TO "
340 PRINT "FOR W = 0 TO 100:
350 PRINT "NAME: W; "
360 PRINT "WORD: W; "
370 PRINT "SCORES:
380 PRINT ""
***** Listing of Program "TESTS" *****

07/10/83 - D.A. O'Leary

100 PRINT "L."
110 NEXT M
120 FOR I = 1 TO 24
130 IF I = 1 THEN GOSUB 420
140 PRINT
150 GOTO 120
160 IF E = 1 THEN DO 100
170 I = +1
180 INPUT "E, N: "; I
190 GOTO 130
200 IF 2 E = 1 THEN GOSUB 320
210 D = I + 1
220 FOR M = 1 TO 24
230 PRINT "L."
240 PRINT
250 GOTO 130
260 IF E = 1 THEN RETURN
270 PRINT "YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:
280 PRINT "PART 1. SUBROUTINES"
290 PRINT "PART 2. LIBRARY FUNCTIONS"
300 PRINT "PART 3. USER FUNCTIONS"
310 GOSUB 410
320 FOR I = 1 TO 24
330 PRINT I
340 NEXT I
350 RETURN
360 REM ** this routine clears the screen
370 REM **
380 FOR I = 1 TO 24
390 PRINT I
400 NEXT I
410 RETURN
420 REM **
430 REM ** this subroutine is for the response section
440 RETURN
450 REM
460 INPUT "ENTER the letter opposite the correct answer": T
470 RETURN
***** Listing of Program 'TESTS' *****

4220 PRINT
4221 RUN"MENU"
4224 RUN"AWS"
4229 CLOSE
4230 OPEN"I"..."SCORES"
4270 IF EOF THEN STOP
4280 INPUT#1,AX,N
4290 PRINT&N,N
4290 GOTO 4279
***** Listing of Program 'LESCEA' *****

1000 REM ** THIS PROGRAM STARTED ON 15 JUNE 1982
1010 REM ** AUTHOR: CAPTAIN DANNY J. CREAGAN
1020 REM ** TITLE: LESSON 3
1030 REM **
1040 REM **
1050 REM **
1060 REM **
1070 REM **
1080 CLEAR 160
1090 GOSUB 6510
1100 PRINT LESSON: BASIC 3
1110 PRINT "VERSION: 1 AUGUST 82"
1120 PRINT
1130 PRINT "TIME REQUIRED TO COMPLETE LESSON: Less than one hour"
1140 PRINT
1150 PRINT
1160 PRINT "AUTHOR: Capt. Danny J. Creagan"
1170 PRINT "Air Force Institute of Technology"
1180 PRINT
1190 PRINT "OBJECTIVE: To teach the student about string functions"
1200 PRINT "and the Microsoft Editor"
1210 PRINT
1220 PRINT
1230 PRINT
1240 PRINT
1250 GOSUB 5510
1260 GOSUB 5550
1270 PRINT "A: I'm taking this part in its entirety..."
1280 PRINT "B: I wish to review selected areas..."
1290 PRINT "C: I want to go to the second part..."
1300 PRINT "D: I want to return to the Menu..."
1310 PRINT
1320 INPUT "Press either capital A, B, C, or D and then press ENTER..." "
1330 IF $1 = "A" GOTO 7000
1340 IF $1 = "B" GOTO 7020
1350 IF $1 = "C" GOTO 1450
1360 IF $1 = "D" GOTO 1270
1370 GOSUB 1560
1380 GOSUB 2220
1390 GOSUB 2220
1400 GOSUB 4570
1410 GOSUB 4570
1420 GOSUB 4570
1430 GOSUB 4570
1440 GOTO 1150
1450 GOSUB 5210
### Listing of Program 'LESSONS' ###

1400 SUB data
1410 PRINT
1420 PRINT "Please type in the number beside the area you wish"
1430 PRINT "to review. Press 1 through 5 and then press 0 and"
1440 PRINT "press ENTER to return to the Menu."
1450 PRINT
1460 INPUT "What is your choice?": N
1470 IF N = 0 THEN 7000
1480 ON N GOSUB 1520, 1530, 1540, 1550, 1560
1490 GOSUB 550
1500 PRINT
1510 PRINT  
1520 PRINT "INTRODUCTION"
1530 PRINT
1540 PRINT "This is your last lesson! Congratulations! If you have taken"
1550 PRINT "the previous 5 lessons, you should be feeling a little more"
1560 PRINT "comfortable with Microsoft BASIC by now."
1570 PRINT
1580 PRINT "In this lesson we will cover STRING functions first, and then"
1590 PRINT "we will learn about the Microsoft Editor in the second half."
1600 PRINT
1610 PRINT "As we learned before, a BASIC string is one or more alpha-"
1620 PRINT "numeric characters that are treated as a single collection of"
1630 PRINT "data. Using the concepts in this chapter, you can perform"
1640 PRINT "many of the same types of operations on STRINGS that you"
1650 PRINT "can perform on numeric data."
1660 PRINT
1670 PRINT
1680 INPUT "Press ENTER if you would like"
1690 GOSUB 550
1700 PRINT  
1710 PRINT "INTRODUCTION"
1720 PRINT
1730 PRINT "As a small review, you should remember that string data can be"
1740 PRINT "designated in two ways. You can assign your data to a string"
1750 PRINT "variable, or you can enclose the data in quotes. Here are two"
1760 PRINT "examples."
1770 PRINT
1780 PRINT "This is one way to designate a string "CHAP 14"
1790 PRINT "Another way is to put it in a variable and print it "CHAP 14"
1800 PRINT
1810 PRINT "Line 10 prints the string data immediately, and line 20 loads"
1820 PRINT "the variable % with the data. % can then be printed whenever"
1830 PRINT "ever we want it."
1840 PRINT
1850 INPUT "Press ENTER if you would like"
1860 GOSUB 550
1870 PRINT  
1880 PRINT  

---

1520, 1530, 1540, 1550, 1560 refer to different sections of the program:
- 1520: Introduction
- 1530: Review of previous lessons
- 1540: Review of string functions
- 1550: Introduction to Microsoft Editor
- 1560: Conclusion and next steps
***** Listing of Program 'LESStNb' *****

1920 PRINT
1921 PRINT"10 PRINT "CHR$(44)"This is one way to designate a string"CHR$(44):
1940 PRINT"D0 V= "CHR$(44)"And this is another"CHR$(44):$
1950 PRINT"D0 PRINT"
1960 PRINT"10 PRINT V$
1970 PRINT"RUN"
1980 PRINT
1990 PRINT"This is one way to designate a string"$
2000 PRINT
2010 PRINT"And this is another$"$
2020 PRINT
2030 PRINT"Here is another variation of our little program. Note that$"$
2040 PRINT"And this is another is not printed until line 40 is executed."$
2050 PRINT
2060 INPUT"press ENTER":V$
2070 GOSUB 6510
2080 PRINT$
2100 PRINT"INTRODUCTION$"
2090 PRINT
2100 PRINT"Remember, in this lesson, as in all of our lessons, you should$"
2110 PRINT"have either a good BASIC manual handy, or you should have"$
2120 PRINT"an experienced programmer around to help you with difficult$"
2130 PRINT"problems. "$"
2140 PRINT
2150 PRINT"In some of the answers you will need to be sure you use the$"
2160 PRINT"correct case (either uppercase or lowercase), so be sure to"$
2170 PRINT"read all the questions carefully."$
2180 PRINT
2190 PRINT"Get out your manual, or programmer, now, and let's enjoy BASIC$"
2200 PRINT
2210 INPUT"press ENTER":V$
2220 GOSUB 6510
2230 PRINT$
2240 PRINT"STRING ASSIGNMENT$"
2250 PRINT
2260 PRINT"As we showed you in the introduction, you assign strings to$"
2270 PRINT"a variable and then you can print the variable anywhere in the"$
2280 PRINT"program. That makes it easier to write long program lines."$
2290 PRINT"Because you don't have to keep typing in the text every time$"
2300 PRINT"you want to use the string data."$
2310 PRINT
2320 PRINT"You can assign data to strings using any of the statements we$"
2330 PRINT"used to assign numeric data to numeric variables. LET, READ$"
2340 PRINT"and INPUT are all used with string assignment (LET is optional$"
2350 PRINT"just as it is with numeric data)."$
2360 PRINT
2370 INPUT"press ENTER":V$
2380 GOSUB 6510
Listing of Program "LESSON6"

1280 PRINT "Are these statements legal" (assume the program is just for"
1290 PRINT "demonstration, and that $ is blank")
1300 PRINT
1310 PRINT "READ A$"
1320 INPUT "INPUT A$"
1330 LET A$ = "$"
1340 PRINT "$ = " + CHR$(44) + "NOW IS THE TIME" + CHR$(44)
1350 PRINT $ = "NOW IS THE TIME" + CHR$(44)
1360 PRINT
1370 PRINT "No. the LET statement in line 20 is illegal"
1380 PRINT "No. the string assignment in line 40 is illegal"
1390 PRINT "No. you cannot read data into a string (line 10 is bad)"
1400 PRINT "yes, all statements are legal"
1410 PRINT
1420 INPUT "ENTER the letter opposite the correct answer": T$%
1430 PRINT
1440 IF T$ = "D" THEN GOTO 2570
1450 PRINT "WELCOME - all these assignments are legal"
1460 GOTO 2580
1470 PRINT "CORRECT - SUPER"
1480 PRINT
1490 A$ = "$" + "$" + CHR$(44) + "NOW IS THE TIME" + CHR$(44)
1500 PRINT "press ENTER": T$
1510 GOSUB 6510:
1520 PRINT "String Assignment"
1530 PRINT
1540 GOSUB 9730
1550 PRINT "RUN"
1560 PRINT
1570 PRINT "THE GRIND"
1580 PRINT "$'s COMING"
1590 PRINT
1600 PRINT "Notice that $ was converted to $, and all the data was"
1610 PRINT "printed out by using just $ in print statements."
1620 PRINT
1630 INPUT "press ENTER": T$
1640 GOSUB 9510
1650 PRINT "String Assignment"
1660 PRINT
1670 GOSUB 9790
1680 PRINT
1690 PRINT "Notice the dollar sign is always included with a string"
1700 PRINT "variable. The dollar sign tells the computer to treat the"
1710 PRINT "variable as a string instead of as a numeric. Also, whenever"
1720 PRINT "you assign data to a string, it must either be another string"
## Listing of Program 'LESSON6' ##

```
2840 PRINT
2850 INPUT press ENTER' T$
2860 GOSUB 6510
2870 PRINT "Assign NOW IS THE TIME to a string variable called NNS"*
2880 PRINT 'and use line number 10 as your statement number.'
2890 PRINT
2900 PRINT "Put one blank between terms."
2910 PRINT
2920 LINE INPUT 'ENTER your answer now? "$'
2930 S$ = "$CHR$(34)" & NOW IS THE TIME" & CHR$(34)
2940 H$ = "$CHR$(34)" & NOW IS THE TIME" & CHR$(34)
2950 PRINT
2960 IF T$ = S$ OR T$ = H$ THEN GOTO 2990
2970 PRINT "WRONG - the correct answer is $CHR$(34)" & NOW IS THE TIME" & CHR$(34)
2980 PRINT "YOU COULD HAVE ALSO ANSWERED $CHR$(34) NOW IS THE TIME" & CHR$(34)"
2990 GOTO 3200
3000 PRINT "CORRECT"*
3010 PRINT*
3020 INPUT press ENTER' T$
3030 GOSUB 6510
3040 PRINT "You can also INPUT string data from an external file."
3050 PRINT "the following program assumes that a file named TEST was"
3060 PRINT "previously created on disk"*
3070 PRINT
3080 PRINT "OPEN $CHR$(34)" & "$CHR$(34)" & "$CHR$(34)" & "$CHR$(34)" & "$CHR$(34)"
3090 PRINT "$CHR$(34)" & "$CHR$(34)"
3100 PRINT "IF EOF(1) THEN END"
3110 PRINT "INPUT..}"*
3120 PRINT "PRINT #1 PRINT A$
3130 PRINT $5 GOTO 20$
3140 PRINT
3150 PRINT "A program such as this is used to read in your name when you"
3160 PRINT "take your test at the end of each lesson. The original is"
3170 PRINT "enhanced a little, but the BASIC idea is the same."*
3180 PRINT
3190 INPUT press ENTER' T$
3200 GOSUB 6510
3210 IF T$ = "$" THEN GOTO 2220
3220 RETURN
3230 PRINT "String ARRAYS"*
3240 PRINT
3250 PRINT "You can assign string data to arrays in the same way as you"
3260 PRINT "assign numeric data to arrays. Nearly all the rules are the"
3270 PRINT "same. The following is an example."
3280 PRINT
3290 PRINT
3300 GOSUB 6510
```

---

129
**** Listing of Program 'LESSON6' ****

770 PRINT
7710 INPUT "Press ENTER"; Tn
7720 GOSUB 6510
7730 GOSUB 6870
7740 PRINT
7750 PRINT "Notice the CLEAR statement. Remember that you normally have"
7760 PRINT "only 50 - 100 characters of string space available, and if you"
7770 PRINT "are going to need more, you need to tell the computer. Also."
7780 PRINT "note the DIM statement - we need declare our array size if it"
7790 PRINT "is over 10"
780 PRINT
7810 INPUT "Press ENTER"; Tn
7820 GOSUB 6510
7830 GOSUB 6870
7840 PRINT
7850 PRINT "The variable X acts as a counter to reference the proper pocket of"
7860 PRINT "the string array. The string array is referenced exactly like"
7870 PRINT "the numeric array. Note that this program will only read in"
7880 PRINT "the data. If you want to print it out, you will have to add"
7890 PRINT "some more statements on the bottom of the program."
790 PRINT
7910 INPUT "Press ENTER"; Tn
7920 GOSUB 6510
7930 PRINT
7940 PRINT "What is the CLEAR statement for in BASIC?"
7950 PRINT
7960 PRINT "A To clear extra number space for the computer"
7970 PRINT "B To clear extra string space"
7980 PRINT "C To zeroize all number variables"
7990 PRINT "D To clear the screen"
8000 PRINT "E To help the programmer understand more clearly"
8010 PRINT
8020 INPUT "Enter the letter opposite the correct answer"; Tn
8030 PRINT
8040 IF Tn = "B" THEN GOTO 3680
8050 PRINT "Wrong - the correct answer is B"
8060 PRINT "this is an important concept, you may wish to review"
8070 PRINT "lesson 3 before you go to the next section."
8080 GOTO 3680
8090 PRINT "Correct"
8100 INPUT "Press ENTER"; Tn
8110 GOSUB 6510
8120 PRINT
8130 INPUT "Press ENTER"; Tn
8140 GOSUB 6510
8150 PRINT
8160 INPUT "Press ENTER"; Tn
8170 GOSUB 6870
8180 PRINT
8190 PRINT
8200 PRINT
8210 PRINT
8220 PRINT
8230 PRINT
8240 PRINT
8250 PRINT
8260 PRINT
8270 PRINT
8280 PRINT
8290 PRINT
8300 PRINT
8310 PRINT
8320 PRINT
8330 PRINT
8340 PRINT
8350 PRINT
8360 PRINT
8370 PRINT
8380 PRINT
8390 PRINT
8400 PRINT
8410 PRINT
8420 PRINT
8430 PRINT
8440 PRINT
8450 PRINT
8460 PRINT
8470 PRINT
8480 PRINT
8490 PRINT
8500 PRINT
8510 PRINT
8520 PRINT
8530 PRINT
8540 PRINT
8550 PRINT
8560 PRINT
8570 PRINT
8580 PRINT
8590 PRINT
8600 PRINT
8610 PRINT
8620 PRINT
8630 PRINT
8640 PRINT
8650 PRINT
8660 PRINT
8670 PRINT
8680 PRINT
8690 PRINT
8700 PRINT
8710 PRINT
8720 PRINT
8730 PRINT
8740 PRINT
8750 PRINT
229
*LISTING OF PROGRAM 'LESSON 6'*

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 3860
3850 PRINT "CORRECT."
3860 PRINT
3870 INPUT "press ENTER": T$
3880 GOSUB 3510
3890 PRINT "How would you find out what was in the fifth pocket of the"*
3900 PRINT "single dimension array A$(1)?"
3910 PRINT
3920 A$ = PRINT A$(X)
3930 PRINT A$ + PRINT A$(5)
3940 PRINT C: PRINT A$
3950 PRINT D: READ A$(X)
3960 PRINT
3970 INPUT "PRESS the letter opposite the correct answer": T$
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 3510
4080 RETURN
4090 GOSUB 3510
4100 PRINT "Concatenation."
4110 PRINT
4120 PRINT "You may link two strings together by using the *plus* symbol."
4130 PRINT "For example:"*
4140 PRINT
4150 PRINT A$ = "CHR$(34)""CHR$(34)"
4160 PRINT BS = "CHR$(34)""Some""CHR$(34)"
4170 PRINT CS = BS * A$
4180 PRINT "You could have said "PRINT BS * A$": too."
4190 PRINT "RUN"
4200 PRINT

230
4210 PRINT "Somewhere"
4220 PRINT "In this case, the '+' symbol served to 'add' the two strings."
4230 PRINT "The correct answer is C."
4240 INPUT "Press ENTER to try another string."
4250 GOSUB 4510
4260 PRINT "What is the output of the following program?"
4270 PRINT "A FLASH"
4280 PRINT "B FLASH"
4290 PRINT "DANCE"
4300 PRINT "A DANCE B"
4310 INPUT "Enter the letter opposite the correct answer."
4320 IF T$ = "C" THEN GOTO 4440
4330 PRINT "The correct answer is C."
4340 GOTO 4450
4350 PRINT "Correct!"
4360 INPUT "Press ENTER to try another string."
4370 GOSUB 4510
4380 PRINT "The correct answer is C."
4390 PRINT "A DANCE B"
4400 PRINT "B FLASH"
4410 IF T$ = "B" THEN GOTO 4490
4420 PRINT "Correct!"
4430 PRINT "Correct!"
4440 GOTO 4550
4450 RETURN
4460 PRINT "Enter for some examples of string functions."
4470 PRINT "String Functions"
4480 PRINT "As with arithmetic functions, there are STRING functions."
4490 PRINT "STRING functions are used to manipulate or explore the contents of a string. In the next screen there are several examples of STRING functions. We will go over several of these, but you'll 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."
4500 PRINT "As you saw in lesson 5, STRING functions can be used as part of USER functions."
4510 GOSUB 4510
4520 PRINT "As you saw in lesson 5, STRING functions can be used as part of USER functions."
4530 PRINT "Press ENTER for some examples of string functions."
4610 GOSUB S510
4620 PRINT "String Functions"
4700 PRINT "1) ASC(string)"
4710 PRINT "2) CHR$(int)"
4720 PRINT "3) FRE(string)"
4730 PRINT "4) INKEY$"
4740 PRINT "5) LEN(string)"
4750 PRINT "6) MID$(string,position,length)"
4760 PRINT "7) RIGHT$(string,length)"
4770 PRINT "8) LEFT$(string,length)"
4780 PRINT "At first glance, these functions look like a lot of GREEK, in fact, they look pretty bad at second glance. However, they are really quite simple 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." 
4790 PRINT "We will go over examples of a couple to help you catch on."
4800 PRINT "Press ENTER to continue."
4810 GOSUB S510
4820 PRINT "Is the following statement TRUE or FALSE?"
4830 PRINT "String functions are used to manipulate data within string" 
4840 PRINT "variables."
4850 PRINT "A TRUE"
4860 PRINT "B FALSE"
4870 PRINT "ENTER the letter opposite the correct answer:"
4880 IF $A = "A" THEN GOTO 4980
4940 PRINT "WRONG - string functions ARE used to manipulate string variables."
4950 GOTO 4990
4960 PRINT "CORRECT"
4970 GOTO 4990
5000 PRINT "Press ENTER to continue."
5010 GOSUB S510
5020 PRINT "String Functions"
5030 PRINT
5040 GOSUB S490
5350 PRINT "TO PRINT ASC(A)$:"
5050 PRINT 
5060 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 ASCII value of 'A'?"
5070 PRINT "You should have rounded it to be 65 decimal."
5080 PRINT "When your computer writes data files to disk, it usually writes
***** Listing of Program 'LESSONo' *****

5120 PRINT "them in ASCII code, one letter at a time. This function has"
5140 PRINT "you when you are trying to convert characters to their number"
5150 PRINT "equivalent."
5160 PRINT
5170 INPUT "press ENTER:*$%"
5180 GOSUB 5510
5190 PRINT "String Functions"
5200 PRINT
5210 PRINT "PRINT CHR$(17):"
5220 PRINT
5230 PRINT "CHR$(n) returns the opposite of the ASCII string function."
5240 PRINT "It returns a character equivalent of decimal 77, which is 'M'."
5250 PRINT
5260 INPUT "press ENTER:*$%"
5270 GOSUB 5510
5280 PRINT "What is the output of the following program?"
5290 PRINT
5300 PRINT 10 PRINT ASC("CHR$ (A) " + "CHR$ (B) " + ""
5310 PRINT 20 PRINT CHR$(17)
5320 PRINT
5330 PRINT A" 
5340 PRINT "64"
5350 PRINT "93"
5360 PRINT "B"
5370 PRINT "A" 
5380 PRINT "B"
5390 PRINT "C"
5400 PRINT
5410 INPUT "ENTER the letter opposite the correct answer:*$%"
5420 PRINT
5430 IF T$ = "B" THEN GOTO 5480
5440 PRINT "WRONG - the correct answer is A!"
5450 PRINT "be sure to use your manual!"
5460 GOTO 5470
5470 GOTO 5490
5480 PRINT "CORRECT - GREAT!"
5490 PRINT
5500 INPUT "press ENTER:*$%"
5510 GOSUB 5510
5520 PRINT "String Functions"
5530 PRINT
5540 GOSUB 5780
5550 PRINT "20 PRINT LEN(A)"
5560 PRINT
5570 PRINT "LEN(string): is a function that returns the length of the string"
5580 PRINT "that is in parenthesis. In this case it should return 17."
```plaintext
520 PRINT
530 PRINT"What is the value of LEN("CHR$(14)"TOM SWIFT"CHR$(14))""
540 PRINT
550 PRINT'A 11"
560 PRINT'B 3"
570 PRINT'C 1"
580 PRINT'D 9"
590 PRINT
600 INPUT"ENTER the letter opposite the correct answer: 1"
610 IF T$ = "A" THEN GOTO 5720
620 PRINT"WRONG - the correct answer is D"
630 GOTO 5730
640 PRINT"CORRECT"
650 PRINT
660 INPUT"press ENTER: 1"
670 GOSUB 6100
680 PRINT"How you have to do some work for yourself. What will be the"
690 PRINT"output of the following program?"
700 PRINT
710 G0SUB 6000
720 PRINT"20 PRINT LEFT$(A,B)"
730 PRINT
740 PRINT"A"
750 PRINT"B"
760 PRINT"C"
770 PRINT"D"
780 PRINT"Nothing will be output"
790 PRINT
800 INPUT"ENTER the letter opposite the correct answer: 1"
810 PRINT
820 IF T$ = "A" THEN GOTO 5920
830 PRINT"WRONG - the correct answer is A"
840 PRINT
850 GOTO 5940
860 PRINT"CORRECT"
870 PRINT
880 INPUT"press ENTER: 1"
890 GOSUB 5910
900 GOSUB 5990
910 PRINT"20 PRINT MID$(A,B,C)"
920 PRINT
930 INPUT"ENTER the output of this program: 1"
940 PRINT
950 IF T$ = "aching" THEN GOTO 5720
960 PRINT"WRONG - the correct answer is aching"
970 GOTO 5730
```
***** Listing of Program "LESSONS" *****

120 PRINT"CORRECT - I'm glad to see you use the book!"
130 PRINT
140 INPUT"press ENTER":TS
150 GOSUB 3510
170 GOSUB 3390
190 PRINT"PRINT RIGHT$&%4:"
110 PRINT
120 INPUT"ENTER the output of this program":TS
130 PRINT
140 IF TS = "gcess" THEN GOTO 1170
150 PRINT"WRONG - the correct answer is gcess"
160 GOTO 1180
170 PRINT"CORRECT - good job"
180 PRINT
190 INPUT"press ENTER":TS
200 GOSUB 3510
210 PRINT String Functions
220 PRINT
230 PRINT"Here's an interesting function. It's called INKEY$ and it"
240 PRINT"strokes your keyboard ONCE and if a key is depressed, it"
250 PRINT"returns the character that was pressed. Here is an example of"
260 PRINT"how to use it."
270 PRINT
280 PRINT"IF INKEY$ = "CHR$(34)*CHR$(34): THEN END"
290 PRINT"RETURN"
300 PRINT
310 PRINT"If you type this program in exactly as shown, and Run it, it"
320 PRINT"will keep running until you press the 'B' key. Try it when you"
330 PRINT"are done here."
340 PRINT
350 INPUT"press ENTER":TS
360 GOSUB 3300
370 IF TS = "9" THEN GOTO 4500
380 RETURN
390 GOSUB 3510
400 PRINT"You have finished the first part of lesson 5. If you wish to"
410 PRINT"review this part, type in 'R'. If you want to continue to the"
420 PRINT"next half, type in 'C'"
430 PRINT
440 INPUT"ENTER an R or a C":TS
450 IF TS = "R" THEN Run
460 IF TS = "C" THEN GOTO 5190
470 GOTO 0700
480 REM **
490 REM ** This subroutine clears the screen on any terminal
500 REM **
s510 FOR I = 1 TO 24
s520 PRINT
s530 NEXT I
s540 RETURN
s550 PRINT* "LESSON 6"
s560 PRINT
s570 PRINT"This is the first part of a two part lesson"
s580 PRINT"It is divided into the following sections."
s590 PRINT
s600 PRINT*1) Introduction 4) Concatenation"
s610 PRINT*2) String Assignment 5) String Functions"
s620 PRINT*3) String ARRAYS"
s630 PRINT
s640 PRINT
s650 PRINT
s660 GOSUB 6510
s670 PRINT"Which do you wish to do?"
s680 PRINT
s690 PRINT"A) Continue on"
s700 PRINT"B) Review this section again"
s710 PRINT
s720 INPUT"Press the letter opposite the correct answer and press ENTER":I$
s730 IF I$ = "A" THEN I$ = "B" THEN RETURN
s740 GOTO 660
s750 REM **
s760 REM ** subroutine for string assignment example
s770 REM **
s780 PRINT*10 READ A$,B$"
s790 PRINT*20 PRINT A$"
s800 PRINT*30 A$ = B$"
s810 PRINT*40 PRINT A$"
s820 PRINTED DATA "CHR$(1:"THE GRINCH"CHR$(24)"."CHR$(34)"IS COMING"CHR$(34)"
s830 RETURN
s840 REM **
s850 REM ** subroutine for arrays example
s860 REM **
s870 PRINT*10 CLEAR 2000"
s880 PRINT*20 DIM A$(100)"
s890 PRINT*30 I = 0"
s900 PRINT*40 I = 1"
s910 PRINT*50 INPUT "CHR$(I):"ENTER up to 25 strings, ENTER 'END' to stop"CHR$(34):"A$""
s920 PRINT*50 IF A$ = "CHR$(34):END"CHR$(34)" THEN END"
s930 PRINT*40 GOTO 40"
s940 RETURN
s950 REM **
s960 REM ** this is subroutine for STRING FUNCTIONS
***** Listing of Program 'LESSON6' *****

07/13/82 - 04:14:15

6970 REM **
6980 PRINT "AS = " + CHR$(0) + "My aching fingers" + CHR$(0)
6990 RETURN
7000 REM **
7010 REM **
7020 RUN "MENU"
7030 PRINT
7040 PRINT
7050 PRINT "Going to the next part, please stand by"
7060 RUN "LESSON6A"
7070 END
**** Listing of Program 'LESSON64' ****

720 30SUB 21000
720 30SUB 22000

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.": $  
920 IF $ = "C" GOTO 49000
922 IF $ = "D" GOTO 48000
923 IF $ = "B" GOTO 1000
940 IF $ = "A" GOTO 890
920 30SUB 2000
955 30SUB 2000
960 30SUB 4000
975 30SUB 8000
976 30SUB 7000
978 30SUB 8000
979 30SUB 10000
980 30SUB 21000
990 GOTO 47000
1000 30SUB 21000
1002 30SUB 5000
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": $  
1040 IF $ = "C" GOTO 46000
1050 IF $ = "B" GOTO 48000
1060 IF $ = "D" GOTO 1000
1070 GOTO 1000
1080 30SUB 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
The object of this lesson 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.

The editor is line oriented, meaning that you operate on the line at a time (and not on a screen of data like you do in a word processor).

Throughout this lesson, we will be using one special line to illustrate all the commands. That line is listed below:

```
SET UP FOR 1 = set UP (PRINT 1: NEXT 1)
```

The editor treats a colon entered on a line as a new line. Therefore, on the above line we have three statements. Obviously, there are several errors in the line.

If you are not familiar with the editor, press ENTER. Pressing ENTER from the EDIT mode updates the line, and puts you back in the EDIT mode again.

If you want to edit the editor after you are done editing, then press ENTER from the EDIT mode.
### Listing of Program 'LESSON6A'

100. INPUT 'press ENTER'
101. GOTO 100
102. PRINT 'Starting'
103. PRINT
104. PRINT: 'FOR i = 1 TO 10: PRINT : NEXT i'
105. PRINT
106. PRINT 'EDIT UP'
107. PRINT 'TO'
108. PRINT
109. PRINT 'In this example, we typed in the word EDIT 10, and the '
110. PRINT 'computer cut us in the EDIT mode. Note that the line number '
111. PRINT 'we are editing appeared. The cursor would normally be flashing'
112. PRINT 'just to the right of the line number.'
113. PRINT
114. PRINT
115. INPUT 'press ENTER'
116. GOTO 115
117. PRINT 'The line to edit is:
118. PRINT
119. PRINT 'ENTER the command now:'
120. PRINT
121. IF i = 'EDIT 10' THEN PRINT 'WRONG - the correct answer is EDIT 10. TRY AGAIN'
122. PRINT 'INPUT press ENTER'
123. PRINT
124. PRINT 'YOU are in the EDIT mode. What do you press to edit the'
125. PRINT 'EDIT mode? Press the space bar DO NOT PRESS BREAK or:'
126. PRINT 'CONTROL C'
127. PRINT
128. PRINT 'EDIT UP'
129. PRINT
130. PRINT 'EDIT DOWN'
131. A = 10
132. IF A < 1 THEN GOTO 130
133. IF A > 10 THEN PRINT 'WRONG - you should have pressed ENTER': PRINT 'TRY AGAIN'
134. PRINT 'INPUT press ENTER to start over:'
135. PRINT 'INPUT press ENTER
136. PRINT: 'Next:'
137. PRINT
138. PRINT 'Note how the rest of the line appeared after you pressed ENTER'
139. PRINT 'That's the way it would happen if you were actually doing it.'
140. PRINT
141. INPUT 'press ENTER'
142. PRINT 'SPACEBAR, RE-ENTER'
143. PRINT
144. PRINT 'Move the cursor over the line you are editing. You can...
145. PRINT press the spacebar, and it will go over the character.'
***** Listing of Program 'LESSON' *****

410 PRINT
411 PRINT 'For a start, let's say you entered the edit mode and the cursor is
412 in just on the right of the line number. Go ahead and press the
413 EDIT button until you get to the end of the line.
414 PRINT 'after the line is printed, press spacebar once more to go on.'
415 PRINT
416 DS = "FOR I = 1 TO 10 : PRINT I: NEXT I"
417 I = 1
418 PRINT "I ="
419 PRINT EDIT "10"
420 PRINT "I ="
421 IF A$ = "1" THEN GOTO 416
422 IF I / LEN(A$) THEN GOTO 420
423 IF A$ = CHR$(32) THEN PRINT MID$(DS,1,1): I = I + 1: GOTO 416
424 GOTO 416
425 PRINT
426 PRINT
427 PRINT "GOOD ... So you see how that worked?"
428 PRINT "Normal... your cursor would also be flashing, and you would"
429 PRINT "still be in EDIT mode, with your cursor or the last character."
430 PRINT
431 INPUT 'Press ENTER':$=
432 GOSUB 25050
433 PRINT "RS :"
434 0: SPACEBAR, read space
435 PRINT
436 PRINT "Notice the lowercase 'n' in the title to this section. That's
437 PRINT ' refers to a feature of the editor that allows you to position;
438 PRINT the cursor 'n' spaces to the right. That way you don't have
439 PRINT to count spaces, at your spacebar to get to the 20th character.
440 PRINT or a long line, all you do is press the number of characters
441 PRINT you want to spacebar over and then press the spacebar.
442 PRINT ' in this example, you are in the edit mode and you want to go
443 PRINT over a few spaces.'
444 PRINT
445 PRINT "Press a number (up to 255), and then press the spacebar..."
446 PRINT
447 PRINT EDIT "10"
448 PRINT "I ="
449 PRINT EDIT "10"
450 IF A$ = "1" THEN GOTO 416
451 IF A$ = CHR$(2) AND A$ = "1" THEN GOTO 450
452 IF A$ = CHR$(2) AND LEN(A$) = 1 THEN GOTO 450
453 IF A$ = "A" THEN GOTO 416
454 IF A$ = "1" THEN GOTO 450
455 A$ = A$ + "$"
440: PRINT "MnEM: That was quick... if you want to do it again, ENTER a 'Y'"
441: INPUT "else ENTER an 'N'": TF
442: IF TF = "Y" THEN GOSUB 21000: GOTO 4255
443: IF TF = "N" THEN GOSUB 21000: GOTO 4450
445: GO TO 21000
450: PRINT "You Pressed ENTER":
451: PRINT "A SPACEBAR, nSearch!"
452: GO TO 21000
460: PRINT "if you entered a bigger number that your line length, then"
461: PRINT "the computer just defaulted to the actual line length."
462: PRINT "Also, if you were actually using the Editor, you would stay in*
463: PRINT "the EDIT mode, with the cursor over the nth character, after*
464: PRINT "you pressed the spacebar:"
465: PRINT "OK, we managed to move the spacebar around a little. what if?"
466: PRINT "we wanted to find a specific character in the line, and we*
467: PRINT "weren't quite sure exactly how far down the line it was?"
468: PRINT "In that case we would use the nSearch! feature."
469: PRINT "Now, press ENTER":
470: GOSUB 21000
471: PRINT "A SPACEBAR, nSearch!"
472: PRINT "With the Search feature, right after you type in EDIT Line*
473: PRINT "you can drive the cursor to any letter in the line*
474: PRINT and if the letter is not in the line, then the editor default*
475: PRINT to the end of the line. The editor only searches to the*
476: PRINT "right of the cursor."*
477: PRINT "For example, you just typed in EDIT 12 and you wish to find*
478: PRINT the letter L. All you do is press 'L' and then press L. The* 
479: PRINT 'editor recognizes upper and lower case, so be sure the case is*
480: PRINT 'right. So ahead and do it now. You have search for an*
481: PRINT 'letter. just type 3 first, then type the letter "
482: PRINT "Now, press EDIT":
483: PRINT "A SPACEBAR, nSearch!"
484: IF AS = "C" THEN GOSUB 4450
485: IF AS = "D" THEN GOSUB 4410
486: IF AS = "E" THEN GOSUB 4410
487: IF AS = "F" THEN GOSUB 4450
488: IF AS = "G" THEN GOSUB 4450
489: IF AS = "H" THEN GOSUB 4450
490: IF AS = "I" THEN GOSUB 4450
491: IF AS = "J" THEN GOSUB 4450
492: IF AS = "K" THEN GOSUB 4450
493: IF AS = "L" THEN GOSUB 4450
494: IF AS = "M" THEN GOSUB 4450
495: IF AS = "N" THEN GOSUB 4450
***** Listing of Program 'LESSON6A' *****

4850 N = INST:0:16:
4860 IF N = 0 THEN N = 1:
4870 PRINT LEFT$(O,N-1):
4880 PRINT
4890 PRINT "NOW Another quick one... ENTER 'y' to do again. else ENTER an"
4900 INPUT "N"":I$:
4910 PRINT
4920 IF I$ = "y" THEN PRINT"!Search for a 't': GOTO 4770"
4930 IF I$ = "n" THEN GOTO 4960
4940 JUMBO 21000:
4950 PRINT"
4960 PRINT
4970 PRINT"You will have noticed that the cursor stopped BEFORE the "
4980 PR. 't' character that you were searching for. That is what is"
4990 PRINT"supposed to happen. In addition, if you asked to search for a"
5000 PRINT"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"'t' 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 "CSR" and the cursor would skip over to the R in"
5080 PRINT"EDIT.":
5090 PRINT
5100 INPUT"Press ENTER to\":
5110 INPUT "Go:"
5120 PRINT
5130 PRINT"nSearchI":
5140 PRINT
5150 PRINT"Here is an example of the nSearchI feature."
5160 PRINT
5170 PRINT"This is the first occurrence of 'R' in line 10."
5180 PRINT"EDIT 10"
5190 PRINT"1:Now type CSR to find 2nd occurrence of R in 10"
5200 PRINT
5210 PRINT
5220 PRINT"10 FOR X = 9 TO 7 IF X = 1"
5230 PRINT
5240 PRINT
5250 PRINT"The cursor would stop just before the nth occurrence of R."
5260 PRINT
5270 INPUT"Press ENTER to"
5280 GOTO 4100
5290 PRINT"What would you type to find the 2nd occurrence of the character?"
5300 PRINT"Let's say you are editing, assume you are already in the"
5310 PRINT"EDIT mode."
5320 PRINT

240
***** Listing of Program 'LESSON6A' *****

5290 INPUT "ENTER your answer now: "$;
5300 PRINT
5310 IF T$ = "C57" THEN GOTO 5330
5320 PRINT "WRONG - you should type C57"
5330 GOTO 5240
5330 PRINT "CORRECT - GREAT!"
5340 PRINT
5350 INPUT "Press ENTER to test another character";
5360 GOSUB 40000
5370 IF T$ = "B" THEN GOTO 4000
5380 RETURN
5390 GOSUB 11100
5400 i = 10 FOR k = 1 TO 7 : PRINT k : NEXT i
5410 k = "FOR k = " : s$ = "" : END
5420 PRINT nDelete!
5430 PRINT
5440 PRINT "Hopefully, you are catching on to the way the boys and girls at"
5450 PRINT "Microsoft are doing things, and won't have too much trouble"
5460 PRINT "with this command. It does what it looks like it does. It"
5470 PRINT "deletes characters, one at a time, or 'n' at a time."
5480 PRINT
5490 PRINT "For example, if you are in the EDIT mode for line 10 and you "
5500 PRINT "want to delete the 'NEW' character, then just press D and the "
5510 PRINT "character will be enclosed in exclamation marks. The excla-
5520 PRINT "mation marks indicate that if you don't change things, then the"
5530 PRINT "new line will not have the character in it."
5540 PRINT
5550 INPUT "Press ENTER to test another character";
5560 GOSUB 21900
5570 PRINT "An example of the Delete; command would be:"
5580 PRINT
5590 PRINT "LET";
5600 PRINT "FOR i = 1 TO 7 : PRINT i : NEXT i"
5610 PRINT "EDIT";
5620 PRINT "10"
5630 PRINT "X:
5640 PRINT "ALL we do is put the cursor to the left"
5650 PRINT "OF the two characters and press D twice"
5660 PRINT "and it would look like this"
5670 PRINT "Then we would press ENTER and we would"
5680 PRINT "have this"
5690 PRINT "Study this example and read the appropriate paragraph in your"
5700 PRINT "manual."
5710 PRINT
5720 INPUT "Press ENTER to test another character";
5730 GOSUB 21900

244
***** Listing of Program 'LESSON6A' *****

07:10:30 - 04:28:19

$050 PRINT"n Delete"
$060 PRINT
$070 PRINT $*
$080 PRINT

$090 PRINT"Now it is your turn. You have to EDIT the above line so that"
$100 PRINT"the characters 'sk' are deleted. You must delete them one at a"
$110 PRINT"time (as we showed you in the previous example). First type"
$120 PRINT"the appropriate command for editing line 10, then move the "
$130 PRINT"spacebar over to the appropriate place, then delete the two"
$140 PRINT"offending characters, then press ENTER" (use upper case)"
$150 PRINT

$160 INPUT ENTER the first command now:'$'
$170 PRINT

$180 IF $> "EDIT 1:" THEN PRINT"WRONG - you should type in EDIT 1:PRINT:S0TO 640$
$190 PRINT$>
$200 = 0$

$210 AS INIAB$
$220 IF AS $='"' THEN SOTO 650$
$230 IF AS $='"' THEN SOTO 650$

$240 IF AS $='"' AND $='" THEN PRINT$>

$250 IF AS $='"' AND $='" THEN PRINT$>

$260 IF AS $='"' AND $='" THEN PRINT$>

$270 IF AS $='"' AND $='" THEN PRINT$>

$280 IF AS $='"' AND $='" THEN PRINT$>

$290 IF AS $='"' AND $='" THEN PRINT$>

$300 IF AS $='"' AND $='" THEN PRINT$>

$310 IF AS $='"' AND $='" THEN PRINT$>

$320 IF AS $='"' AND $='" THEN PRINT$>

$330 IF AS $='"' AND $='" THEN PRINT$>

$340 IF AS $='"' AND $='" THEN PRINT$>

$350 IF AS $='"' AND $='" THEN PRINT$>

$360 IF AS $='"' AND $='" THEN PRINT$>

$370 IF AS $='"' AND $='" THEN PRINT$>

$380 IF AS $='"' AND $='" THEN PRINT$>

$390 IF AS $='"' AND $='" THEN PRINT$>

$400 IF AS $='"' AND $='" THEN PRINT$>

$410 IF AS $='"' AND $='" THEN PRINT$>

$420 IF AS $='"' AND $='" THEN PRINT$>

$430 IF AS $='"' AND $='" THEN PRINT$>

$440 IF AS $='"' AND $='" THEN PRINT$>

$450 IF AS $='"' AND $='" THEN PRINT$>

$460 IF AS $='"' AND $='" THEN PRINT$>

$470 IF AS $='"' AND $='" THEN PRINT$>

$480 IF AS $='"' AND $='" THEN PRINT$>

$490 IF AS $='"' AND $='" THEN PRINT$>

$500 IF AS $='"' AND $='" THEN PRINT$>

$510 IF AS $='"' AND $='" THEN PRINT$>

$520 IF AS $='"' AND $='" THEN PRINT$>

$530 IF AS $='"' AND $='" THEN PRINT$>

$540 IF AS $='"' AND $='" THEN PRINT$>

$550 IF AS $='"' AND $='" THEN PRINT$>

$560 IF AS $='"' AND $='" THEN PRINT$>

$570 IF AS $='"' AND $='" THEN PRINT$>

$580 IF AS $='"' AND $='" THEN PRINT$>

$590 IF AS $='"' AND $='" THEN PRINT$>

$600 IF AS $='"' AND $='" THEN PRINT$>

$610 IF AS $='"' AND $='" THEN PRINT$>

$620 IF AS $='"' AND $='" THEN PRINT$>

$630 IF AS $='"' AND $='" THEN PRINT$>

$640 IF AS $='"' AND $='" THEN PRINT$>

$650 PRINTThe line in the computer's memory would now look like this:"PRINT '"FOR $= 1 TO 77 PRINT "$INEXT 2"'

$670 PRINT

$680 PRINT"In the example, we protected you from mistakes by ignoring"
$690 PRINT"some commands, and telling you what you did wrong for others."
$700 PRINT"If you really are editing a line, be sure you press the right"
$710 PRINT"buttons, because you won't get warning messages! (however)."
$720 PRINT"You seldom ruin what you have done, the editor is very for-"
$730 PRINT"tuning. It usually leaves you something, even when you make a"
$740 PRINT"boo-boo."

245
**** Listing of Program 'LESSON6A' ****

1740 PRINT
1750 PRINT "As you may have guessed, the 'n' symbol in the title indicates"
1760 PRINT "that you may delete 'n' characters at a time. If you press a"
1770 PRINT "number and press D, then that is how many characters will be"
1780 PRINT "deleted."
1790 PRINT
1800 INPUT "press ENTER*17$"
1810 GOSUB 11000
1820 PRINT
1830 PRINT
1840 PRINT "If we wanted to delete 's' all once, we could position the"
1850 PRINT "cursor to just before the 's' and type in 20 and both "
1860 PRINT "characters would appear like this: 's'"
1870 PRINT
1880 PRINT "Note the exclamation marks, in this case, are around both"
1890 PRINT "characters. When you get through here, practice with some"
1900 PRINT "lines you have arbitrarily made up. You will see that this"
1910 PRINT "command can be very handy."
1920 PRINT
1930 INPUT "press ENTER*17$"
1940 GOSUB 14000
1950 IF T$ = "B" THEN GOTO 1960
1960 RETURN
1970 GOSUB 11000
1980 PRINT "Xtend line/"
1990 PRINT
2000 PRINT
2010 PRINT "This is one of the easiest, and most useful of the commands"
2020 PRINT "It allows you to start up at the end of a line, just as if you"
2030 PRINT "never pressed ENTER."
2040 PRINT
2050 PRINT "First, you go to EDIT mode, then you press k. When you do, you"
2060 PRINT "will see the whole line displayed, and you can add anything"
2070 PRINT "to the end of it. Now, first, type the command to"
2080 PRINT "get into EDIT mode for line 10, then press k, then type in "
2090 PRINT "anything you want (most micros allow a maximum of 248"
2100 PRINT "characters!). Then press ENTER. Do it now."
2110 PRINT "*40, OS = "FOR X = 107" PRINT X :NEXT I"
2120 INPUT "ENTER the first command*17$"
2130 IF T$ = "EDIT 10" THEN PRINT PRINT 1790 PRINT "WRONG - you should type in EDIT 10 first":PRINT:GOTO 7150
2140 PRINT
2150 PRINT 1790 PRINT 101:
2160 AS = INKEY$!
2170 IF AS = "c" THEN AS = "I"
2180 IF AS = "I" THEN GOTO 1790
2190 PRINT 2160
0010 PRINT "Your new line, which was the sum of the old line 10 plus the"
0020 PRINT "data you typed in, is now this:
0030 PRINT
0040 PRINT "Neat huh? You will find yourself using this command the most."
0050 PRINT
0060 PRINT "As usual, we protected you from making mistakes, by only"
0070 PRINT "allowing you to execute the (tend) command. Remember, you"
0080 PRINT "will have much more freedom if you really are in the editor"
0090 PRINT "mode. In fact, in the real editor, you can use the back arrow"
0100 PRINT "to wipe out the end of the line you are editing, and replace"
0110 PRINT "the old data with new stuff. Be sure to practice this and"
0120 PRINT "you will grow to love it! -- ....well, maybe just like it a lot."
0130 PRINT
0140 INPUT "Press ENTER":T$
0150 GOSUB 40000
0160 IF T$ = "B" THEN GOTO 7000
0170 RETURN
0180 GOSUB 21000
0190 PRINT n(Change) & Insert"
0200 PRINT
0210 PRINT "Now for the meat of this half! We are going to change the"
0220 PRINT "nasty errors in our trial statement, and then insert some"
0230 PRINT "correct figures."
0240 PRINT
0250 G1$ = "FOR X = to77 :PRINT Y :NEXT Z"
0260 PRINT G1$
0270 PRINT
0280 PRINT "Above is our line (with the 's*' characters missing - we"
0290 PRINT "deleted them in the n(Del)seion). Let's say we want"
0300 PRINT "to change 'NEXT Z' to 'NEXT X' and 'PRINT Y' to 'PRINT A'"
0310 PRINT "In addition, we want the value of X to start at 1 in the FOR"
0320 PRINT "NEXT loop i.e. we want to insert a 1 just before the to77"
0330 PRINT
0340 INPUT "Press ENTER to start our EDITing":T$
0350 GOSUB 21000
0360 PRINT n(Change) & Insert"
0370 PRINT
0380 PRINT
0390 PRINT
0400 PRINT "First we'll change 'PRINT Y' to 'PRINT Z'"
0410 PRINT
0420 PRINT "To use the n(Change) command, get into the EDIT mode and "
PRINT "position the cursor to JUST BEFORE the character to be changed"
PRINT "then press 'C' and press the new character, then press ENTER."
PRINT "Try it now. Set to EDITor, SPACE over to just before the Y"
PRINT "and type in 'C', then type in an X we want to swap Y with it:"
PRINT "and finally, press ENTER (use capitals)"
PRINT
PRINT "ENTER the first command at the bottom of the next line"
PRINT *01%
LINE INPUT T$      
IF T$ <> EDIT 10 THEN PRINT "Wrong - you have to type EDIT 10 first":GOTO 3200
PRINT
I = 0
PRINT 10 ";
A$ = INKEYS
IF A$ = "C" THEN GOTO 3240
IF A$ = "C" THEN A$ = "E"
IF A$ = CHR$(32) AND I < 20 THEN I = I + 1: PRINT MID$(A$,1,1):GOTO 3240
IF A$ = "C" AND I < 20 THEN PRINT:PRINT "Wrong - you must press 'C' just before the Y in PRINT"
PRINT "TRY AGAIN":PRINT: INPUT:press ENTER":T$:PRINT:GOTO 3170
IF A$ = "C" AND I = 20 THEN GOTO 3400
GOTO 3240
A$ = INKEYS
IF A$ = "C" THEN GOTO 3400
A$ = INKEYS
IF A$ = "C" THEN GOTO 3400
PRINT "NEXT"
GOTO 3000
PRINT "How's that for class? Remember, we protected you from mistakes."
PRINT "The real editor will do whatever you tell it, even if it is"
PRINT "wrong. But you knew that, didn't you?"
PRINT
PRINT "As with the other commands, the 'n' in nCHANGE designates"
PRINT "how many characters are affected by the command. If you want"
PRINT "to change 10 characters, then you would type '10C' in the EDIT"
PRINT "mode, and you would then HAVE to change the next 10 characters."
3680 PRINT "What would you type if you were in the EDIT mode and your "
3690 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": TS
3730 PRINT
3740 IF TS <> "4CXXX" THEN PRINT "Wrong - you should have typed 4CXXX": PRINT: PRINT: TRY AGAIN: INPUT "Press ENTER": TS: GOTO 3370
3750 PRINT "Great! Now you have the idea!"
3760 PRINT
3770 INPUT "Press ENTER": TS
3780 GOSUB 21000
3790 PRINT "n(Change) & (I)nsert)"
3800 PRINT
3810 PRINT "10 FOR X = 77: PRINT X; NEXT X"
3820 PRINT
3830 PRINT "We would change the 'I' to an 'X' in the same way, but, to"
3840 PRINT "save time, we'll invoke some magic, and change it now so we can"
3850 PRINT "get to the (I)nsert) command: READY: izzzz1ZZZAAAAPP!"
3860 PRINT "<\ POPF >>. There, it's changed now. Look below."
3870 PRINT
3880 PRINT "FOR X = 77: PRINT X; NEXT X"
3890 PRINT
3900 PRINT "How would you like to have THAT editor at your command?"
3910 PRINT "We'll now get to the (I)nsert) command. Remember, we want to"
3920 PRINT "insert a 'I' just before the '777'."
3930 PRINT
3940 PRINT "Press ENTER for the Insert example": TS
3950 GOSUB 21000
3960 PRINT "(Change) & (I)nsert)"
3970 PRINT "FOR X = 77: PRINT X; NEXT X"
3980 PRINT
3990 PRINT "How would you like to have THAT editor at your command?"
4000 PRINT "We'll now get to the (I)nsert) command. Remember, we want to"
4010 PRINT "insert a 'I' just before the character you want"
4020 PRINT "then place the cursor to just before the character you want"'
4030 PRINT "to insert the data in front of."
4040 PRINT
4050 PRINT "In this case, we get into the EDIT mode, then"
4060 PRINT "We SPACE over to just before the '777' and then we type an 'I'"
4070 PRINT "for (I)nsert). After the 'I' command we want to put in a 'I'"
4080 PRINT "but we COULD type in as many characters we want ...."
4090 PRINT "until we press ENTER. At that time, all our changes are made"
4100 PRINT "and we are returned to the IMMEDIATE mode."
4110 PRINT
4120 INPUT "Press ENTER to start the example": TS

249
***** Listing of Program 'LESSON6A' *****

9170 50SUB 21000
9180 PRINT "Remember, first type EDIT 10, then space over to just before"
9190 PRINT "the 'to?' then type 'I', then type a I. then type ENTER."
9200 PRINT
9210 PRINT "10 "&
9220 PRINT
9230 INPUT "ENTER the first command":$&
9240 IF $ <> "EDIT 10" THEN PRINT "WRONG - you must type EDIT 10 first":PRINT:GOTO 9230
9250 PRINT
9260 PRINT "10 "
9270 I = 0
9280 IF AS = INKEYS THEN GOTO 9270
9290 IF AS = "I" THEN GOTO 9270
9300 IF AS = "I" THEN AS = "1"
9310 IF AS = CHR$(121) AND I < 9 THEN I = I + 1:PRINT MID$(AS, I, 1):GOTO 9270
9320 IF AS = CHR$(120) AND I < 9 THEN PRINT:PRINT "WRONG - don't press ENTER until you are"
9330 THEN PRINT:PRINT "done":PRINT:INPUT "press ENTER to start over":GOTO 9230
9340 IF AS = "1" AND I < 9 THEN PRINT:PRINT "WRONG - don't type I until just before the"
9350 THEN PRINT:INPUT "press ENTER to start again":T$:PRINT:GOTO 9230
9360 IF AS = "I" AND I > 9 THEN GOTO 9400
9370 GOTO 9270
9380 AS = INKEYS
9390 IF AS = "I" THEN GOTO 9490
9400 PRINT AS:
9410 AS = INKEYS
9420 IF AS = CHR$(123) THEN GOTO 9430
9430 PRINT "to?":PRINT X:NEXT I
9440 PRINT
9450 INPUT "Enter a 'Y' to do this again, else ENTER an 'N'":T$&
9460 IF $ = "Y" THEN GOTO 9170
9470 IF $ = "N" THEN GOTO 9470
9480 50SUB 21000
9490 PRINT "Change & (insert):"
9500 PRINT
9510 PRINT "With some of the same magic we invoked before, we will also"
9520 THEN PRINT "put spaces in the proper places of the test line, again."
9530 PRINT "You would normally use your NC or I command to fix up the line"
9540 PRINT "but I feel pretty magical today, so I want to do it."
9550 PRINT
9560 PRINT ":::IIIIIIIIIIIIIIIIIIIIIIIIIIIPOOD :::::..CRAAASSSSHHHH :)inkle"
9570 PRINT "FORGROUND=I:IAISLFUD():+++nnnnnnnnn........dooops:"&
9580 PRINT
9590 PRINT "OK well, we'll leave it up to you to do in your practice"
9600 PRINT "sessions."
9610 PRINT
9620 INPUT "Press ENTER":T$&
**** Listing of Program 'LESSON5A' **** 07/10/83 - 04:29:39

9650  BDSUB 40000
9660  IF TS = "8" THEN GOTO 8000
9670  RETURN
10000  BDSUB 21000
10100  PRINT "You have finished the lesson and you can now take the test."
10200  PRINT "If you wish to review parts of the lesson, ENTER an 'R'"
10300  PRINT "else, if you want to continue to the test ENTER a 'C'"
10400  PRINT
10500  INPUT "ENTER your choice now :R or C";TS
10600  IF TS = "R" THEN RUN
10700  IF TS = "C" THEN GOTO 10000
10700  GOTO 47000
20000  REM **
20100  REM ** This subroutine clears the screen on any terminal
20200  REM **
21000  FOR X = 1 TO 24
21100  PRINT
21200  NEXT X
21300  RETURN
30000  PRINT "LESSON 6"
30100  PRINT
30200  PRINT "This is the second part of a two part lesson"
30300  PRINT "It is divided into the following sections."
30400  PRINT
30500  PRINT 1) Introduction
30600  PRINT 2) Starting (EDIT/exit) 3) Insert line"
30700  PRINT 4) nSpaces, nSearch, nChange, nInsert"
30800  PRINT 5) TEST"
30900  PRINT
31000  PRINT
31100  RETURN
40000  BDSUB 21000
40000  PRINT "Which do you wish to do?"
40100  PRINT
40200  PRINT 1) Continue on"
40300  PRINT 2) Review this section again"
40400  PRINT
40500  INPUT "Please enter the letter opposite the correct answer and press ENTER":TS
40600  IF TS = "A" OR TS = "B" THEN RETURN
40700  GOTO 40000
47000  PRINT
47100  PRINT "Going to test. Please wait one moment."
47200  RUN "TEST."
48000  RUN "MENU"
49000  REM "LESSOND"
50000  END
**** Listing of Program 'TESTa' ****

1000 REM **
1010 REM ** LESSON: TESTa  VERSION: 1 AUG 82
1020 REM ** AUTHOR: CAPT DAN GREASHAN
1030 REM ** AIR FORCE INSTITUTE OF TECHNOLOGY
1040 REM **
1050 REM ** VARIABLES:
1060 REM ** N$: i = NAMES ARRAY, USED TO READ IN SED-
1070 REM ** UENTIAL NAMES, AND TO WRITE OUT
1080 REM ** UPDATE NAMES.
1090 REM ** S(i) = SCORES ARRAY - USED TO READ AND
1100 REM ** WRITE SCORES
1110 REM ** Z(i) = ARRAY TO KEEP TRACK OF NUMBER OF
1120 REM ** CORRECT ANSWERS. IF AN ARRAY
1130 REM ** ELEMENT EQUALS 1, THE ANSWER WAS
1140 REM ** CORRECT
1150 REM **
1160 CLEAR 4000
1170 GOSUB #110
1180 DIM N$:1000
1190 DIM i():100
1200 DIM z():100
1210 PRINT "FINAL TEST (lesson a:"
1220 PRINT
1230 PRINT "This test consists of 10 questions, you must get 70 percent."
1240 PRINT "If you answer correctly, you will 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 #410
1350 PRINT
1360 PRINT "A) Multi Statement lines are not allowed"
1370 PRINT "B) The assignment of values between A$ and B$ are not valid"
1380 PRINT "C) The strings were not initialized"
1390 PRINT "D) Nothing"
1400 PRINT
1410 GOSUB #430
1420 PRINT
1430 IF "$ = 'D'" THEN GOTO 145
1440 PRINT "Wrong - the correct answer is D"
1450 PRINT "See part 1. String Functions"
***** Listing of Program 'TEST6' *****

1460 GOTO 1470
1470 PRINT "CORRECT"
1480 Q(I) = 1
1490 PRINT
1500 INPUT "Press ENTER":$1
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":$1
1580 PRINT
1590 IF S1 = "$H" THEN GOTO 1640
1600 PRINT "WRONG - the correct answer is "$H"
1610 PRINT * If AS = "Hi" and BS = AS, then the left character"
1620 PRINT * of BS is an H. See part 1.*
1630 GOTO 1660
1640 PRINT "CORRECT"
1650 Q(2) = 1
1660 PRINT
1670 INPUT "Press ENTER":$1
1680 GOSUB 4110
1690 PRINT
1700 PRINT "What is wrong with the following program.""
1710 PRINT
1720 PRINT "FOR X = 1 TO 20"
1730 PRINT "A$(X) = "CHR$(34) + "CHR$(34)"
1740 PRINT "NEXT X"
1750 PRINT
1760 PRINT "The array is not dimensioned properly."
1770 PRINT "You cannot address a single dimensioned array with a loop."
1780 PRINT "The "$ should not be enclosed in quotes."
1790 PRINT "Nothing."
1800 PRINT
1810 GOSUB 4230
1820 PRINT
1830 IF S1 = "$M" THEN GOTO 1860
1840 PRINT "WRONG - the answer is A! it should be dimensioned to 20."
1850 PRINT "See part 1. String Arrays."
1860 GOTO 1880
1870 PRINT "CORRECT"
1880 GOSUB 4110
1890 PRINT
1900 READ $1
1910 PRINT "What is the output of the following program?"
Listing of Program 'TEST6'

1920 PRINT
1930 GOSUB 4200
1940 PRINT"20 B$ = A$ + B$"
1950 PRINT
1960 INPUT"ENTER your answer EXACTLY as it would appear":T$
1970 PRINT
1980 IF T$ = "Hi": 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: = 1
2150 PRINT
2160 INPUT"press ENTER":T$
2170 GOSUB 4110
2180 PRINT"What is the output of the following program?"
2190 PRINT
2200 PRINT"10 A$ = "&CHR$(54)"*SOMETHING*CHR$(54):1$:A$ = MID$(A$,1,.4):PRINT A$"
2210 PRINT
2220 INPUT"ENTER your answer EXACTLY as it would appear":T$
2230 PRINT
2240 IF T$ = "SOME": THEN GOTO 2180
2250 PRINT"WRONG - the correct answer is SOME"
2260 PRINT" See your BASIC manual."
2270 GOTO 2200
2280 PRINT"CORRECT"
2290 Q: = 1
2300 PRINT
2310 INPUT"press ENTER":T$
2320 GOSUB 4110
2330 PRINT"What is the command you would enter to edit line number 50 of?"
2340 PRINT"a program?"
2350 PRINT
2360 INPUT"ENTER your answer EXACTLY as it would appear":T$
2370 PRINT
2380 IF T$ = "edit 50": THEN GOTO 2330
2390 IF T$ = "EDIT 50": THEN GOTO 2330
2400 PRINT"WRONG - the correct answer is EDIT 50"
2410 PRINT" See part 2, EDIT"
2420 GOTO 2300
2430 PRINT"CORRECT"
2440 Q: = 1
2450 PRINT
2460 INPUT"press ENTER":T$
2470 GOSUB 4110
***** Listing of Program 'TEST6' *****

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" ZCR"
2440 PRINT"B" ZCR"
2450 PRINT"C" ZCR"
2460 PRINT"D" ZCR"
2470 PRINT"E" spacebar R"
2480 PRINT
2490 GOSUB 4180
2500 PRINT
2510 IF T$ = "A" THEN GOTO 2550
2520 PRINT"WRONG - the correct answer is A"
2530 PRINT" See part 2, nSearch"
2540 GOTO 2570
2550 PRINT"CORRECT"
2560 Q(7) = 1
2570 PRINT
2580 INPUT"press ENTER":T$
2590 GOSUB 4110
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"ENTER the command now":T$
2660 PRINT
2670 IF T$ = "I" OR T$ = ":" THEN GOTO 2710
2680 PRINT"WRONG - the correct answer is I"
2690 PRINT" See Part 2, Insert"
2700 GOTO 2770
2710 PRINT"CORRECT"
2720 Q(7) = 1
2730 PRINT
2740 INPUT"press ENTER":T$
2750 GOSUB 4110
2760 PRINT"Assume you are in the EDIT mode"
2770 PRINT
2780 PRINT"What is the command you would use to drive the cursor to the"
2790 PRINT"end of the line you are currently editing. The command is"
2800 PRINT"one letter long."
2810 PRINT
2820 INPUT"ENTER the command now":T$
2830 PRINT

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

07/10/83 - 04:48:34

2840 IF I$ = "I" OR I$ = "a" THEN GOTO 2880
2850 PRINT"Wrong - the correct answer is I"
2860 PRINT"See part 2. (Extends)"
2870 GOTO 2900
2880 PRINT"Correct"
2890 Q!1 = 1
2900 PRINT
2910 INPUT"Press enter: ITS"
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 21:10 = 1
3020 GOSUB 4110
3030 PRINT
3040 IF I$ = "S" THEN PRINT"Dont blame me ... I'm only following orders"
3045 PRINT"Press enter: ITS"
3050 GOTO 3120
3055 IF I$ = "C" THEN PRINT"That won't help the National Debt very much"
3060 PRINT"Press enter: ITS" GOTO 3120
3065 IF I$ = "B" THEN PRINT"Don't try to butter me up. I know you're into overselling"
3070 PRINT"Press enter: ITS"
3080 GOTO 3120
3085 IF I$ = "A" THEN PRINT"Obviously we have failed to communicate. I'm reporting you to: F.B.I.
3090 FOR tax evasion and mail fraud."
3095 PRINT"Try again"
3100 PRINT"Press enter: ITS" GOTO 3120
3105 PRINT"Congratulations, you are one of the few who selected an answer"
3110 PRINT"Press enter: ITS" GOTO 3120
3115 PRINT"That wasn't listed. Were you ever an extra for the MUPPET SHOW?"
3120 PRINT
3125 INPUT"Press enter: ITS"
3130 GOSUB 4110
3135 PRINT"Obviously, you get automatic credit for the last question."
3140 PRINT
3145 PRINT"It was nice doing business with you. So long"
3150 PRINT
3155 INPUT"Press enter: ITS"
3160 GOSUB 4110
3170 FOR I = 1 TO 10
3180 Y = RND
3190 NEXT I
3200 PRINT"You have finished the test. Out of 10 possible correct answers"
3210 PRINT"You scored "Y;"
3220 PRINT
3230 IF Y > 5 THEN PRINT"YOU HAVE PASSED"
**** Listing of Program "TEST6" ****

1240 SUB 3920
1270 IF Y = "A" THEN GOTO 3950
1280 PRINT"YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS"
1290 PRINT
1300 PRINT"YOU SHOULD RETAKE LESSON 5"
1310 PRINT
1320 PRINT"You will be returned to the Menu."
1330 PRINT
1340 GOTO 4250
1350 PRINT
1360 PRINT"Do you want your score recorded on a permanent file?"
1370 PRINT
1380 PRINT"A YES"
1390 PRINT"B NO"
1400 PRINT
1410 INPUT"Which?"A$1
1420 IF A$ = "B" THEN GOTO 3720
1430 GOTO 435 SUB 4110
1440 PRINT"To record your score, we must know your name. Therefore, surprisingly, we need your name. If your name is not unique among the students likely to take this test, please contact your test monitor for an identifying word that will make you unique. Then enter that word below."
1450 PRINT
1460 PRINT"If you have already entered a score previously, be sure to enter the same name you used before. Use all capitals."
1470 PRINT
1480 PRINT
1490 INPUT"ENTER your name or name now is:"A$
1500 OPEN#1,.1"SCORES"
1510 IF A$ = "0" THEN GOTO 3520
1520 PRINT"PLEASE ENTER YOUR NAME分"
1530 INPUT#1,.1$A$
1540 IF IN$A$ = "YES" THEN GOTO 3720
1550 GOTO 4450
1560 CLOSE
1570 IF A$ = "YES" THEN GOTO 3520
1580 PRINT#1,.1"SCORES"
1590 FOR W = 1 TO 10
1600 PRINT#1,.1$W
1610 PRINT#1,.1$W
1620 NEXT W
1630 PRINT
1640 PRINT
1650 PRINT

165 2
***** Listing of Program "TEST2" *****

1 DO SUB (410)
20 PRINT "You are FINISHED WITH THIS COURSE " "
30 PRINT
40 PRINT "It was GREAT having you as a student - THANKS!"
50 PRINT "When you press ENTER you will be sent to the Menu from"
60 PRINT "where you can review other lessons or quit"
70 PRINT
80 PRINT "For homework, you may wish to change your inventory program"
90 PRINT "so that it will handle string data. That way you can include"
100 PRINT "the names of your furniture in your file. However, we leave"
110 PRINT "that up to you. Hasta Luego!"
120 PRINT
130 GOTO 410
140 END
150 IF EOF(1) THEN CLOSE:GOTO 120
160 : = 1
170 INPUT. #1, (i, EAX)
180 GOTO 160
190 OPEN#3:"SCOREDB"
200 FOR A = 1 TO T
210 PRINT#3, "\n"
220 PRINT#4, "\n"
230 NEXT A
240 GOTO 350
250 PRINT
260 GOTO 770
270 IF #15 THEN RETURN
280 PRINT "YOU NEED IMPROVEMENT IN THE FOLLOWING AREAS:" "
290 PRINT
300 IF A1 = 0 THEN PRINT "part 1. STRING ASSIGNMENT, STRING FUNCTIONS"
310 IF A2 = 0 THEN PRINT "part 1. STRING ASSIGNMENT, STRING FUNCTIONS"
320 IF A3 = 0 THEN PRINT "part 1. STRING ARRAYS"
330 IF A4 = 0 THEN PRINT "part 1. CONCATENATION"
340 IF A5 = 0 THEN PRINT "part 1. STRING FUNCTIONS"
350 IF A6 = 0 THEN PRINT "part 1. EDIT"
360 IF A7 = 0 THEN PRINT "part 2. 3Search"
370 IF A8 = 0 THEN PRINT "part 2. Insert"
380 IF A9 = 0 THEN PRINT "part 2. Delete"
390 PRINT
400 INPUT " press ENTER for " "
410 : CLEAR A10
420 RETURN
430 END **
440 REM ** this subroutine clears the screen"
450 CLEAR **
460 FOR I = 1 TO 24
470 PRINT
***** Listing of Program "TEST5" *****

4130 NEXT I
4140 RETURN
4150 REM **
4160 REM ** this subroutine is for the response section
4170 REM
4180 INPUT "ENTER the letter opposite the correct answer"$I%
4190 RETURN
4200 REM **
4210 REM ** subroutine for string assignment/parameters
4220 REM **
4230 PRINT I$: AS = "CHR$(I])+"I$: CS$ = LEFT$(S$,I$); I$: S$%I$: SC$%
4240 RETURN
4250 PRINT
4260 INPUT 'Press ENTER to return to MENU or...
4270 RUN MENU
4280 CLOSE
4290 OPEN 1: "SCORES"
4300 IF EOF(1) THEN STOP
4310 INPUT S$;N
4320 PRINT AS
4330 S070 40...
# 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 2.00.00 for TRSDOS Version 8
Copyright (c) 1982 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 GRIFKEF AFB, NY. I will be in the Aircraft Maintenance area after Oct 85.

press ENTER to continue?

MENU CHOICES

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

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

WHICH NUMBER DO YOU WANT?
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    6) Print Statement
3) Software     7) End & Stop Statement
4) General Information 8) Immediate Mode, NEW
LIST, DELETE

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

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

Introduction
Throughout all your lessons, you should have your BASIC manual handy. If you find yourself stumped by a question, you should look up the answer in the book. If you can't find it after an honest attempt, then make a guess and then go on. You will have an opportunity to review each section again.

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

press ENTER

Introduction

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

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

press ENTER

Which do you wish to do

A Continue on
B Review this section again

press the letter opposite your choice and press ENTER"A
HARDWARE

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

One major piece of hardware is the central processing unit (CPU). The CPU is the computer’s central electronic brain. It performs all of the data operations and contains a storage area called MEMORY which is used for short term data retention during operations.

press ENTER to continue

HARDWARE (cont)

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

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

You ‘talk’ to the computer through units called INPUT devices.

press ENTER to continue

HARDWARE (cont)

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

INPUT and OUTPUT DEVICES provide a physical communication
Link between you and the computer.

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

Press ENTER to continue.

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

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

WRONG – the correct answer is SOFTWARE.

Press ENTER to continue.

What do the letters CPU stand for? CENTRAL PROCESSING UNIT.

CORRECT – now we are learning something.

Press ENTER to continue.

Which of the following can be considered an OUTPUT device?

A) Terminals
B) Tapes
C) Disk
D) All of the above

Which letter do you select? D

CORRECT
Which do you wish to do

A. Continue on
B. Review this section again

Press the letter opposite your choice and press ENTER. A

SOFTWARE

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

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

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

Press ENTER to continue.

SOFTWARE (cont)

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

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

The rules, or grammar, that you use to write your software
are described by the kind of computer language you use.
press ENTER to continue?

SOFTWARE (cont)

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

press ENTER to continue?

It's time for another quiz.
Remember, use only capital letters and don't add unnecessary spaces or words.
press ENTER to continue?

Is a peripheral, such as a line printer, hardware or software?
A Hardware
B Software

Choose A or B - press the letter and then press ENTER.
Which letter? A
CORRECT
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

dress 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, 190, and 50. 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?
The PRINT statement works like this, to find the average of five numbers and write the result on your terminal.

you can use the following BASIC statement:

```
PRINT (19+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?

Why is BASIC a good general programming language to learn?

A It can be used by most students and programmers
B You don't need to know a lot about computers to use it
C It is available on most computers
D All of the above
Press the letter corresponding to the correct answer
Be sure to enter only capital letters

What is your selection? D

CORRECT

press ENTER to continue?

Which do you wish to do

A  Continue on
B  Review this section again

dress 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
**** Listing of Program 'LEWM01/TAT' ****

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

BASIC WORD
- specifies the computer operation to be performed

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

press ENTER to continue?

STATEMENTS and PROGRAMS (cont)

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

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

press ENTER to continue?

STATEMENTS and PROGRAMS (cont)

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

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

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

press ENTER for the questions
A computer program is a series of ---------

A  Verbs
B  Words
C  Statements
D  Synonyms

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

CORRECT

press ENTER to continue?

Which of the following is incorrect?

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

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

What is your choice? B

CORRECT - way to go!

press ENTER for the next question?

Is 350000 a valid statement number in Microsoft BASIC?

N  No
Y  Yes

press the letter beside the correct answer and then press ENTER? Y
Listing of Program 'LESSON1.TIT' 07/11/83 - 09:04:30

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.

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

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

Results in:
1+1
2
press ENTER?
```

```
PRINT (cont)
10 PRINT "1+1"
20 PRINT 1+1

1+1
2

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

press ENTER?
```

```
PRINT (cont)

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

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

Results in:
A  B  B  B

Press ENTER?

A  B  B  B

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.

Press ENTER?

PRINT (cont)

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

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

Press enter for results?

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

```plaintext
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
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963
END & STOP

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

press ENTER for the rest of END & STOP

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

10 X = 2 + 3
20 Y = X / 5
30 STOP
40 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 an 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 \texttt{LESSONG.TXT} *****  

10 \texttt{x = 12 + 5}  
20 \texttt{y = 3 + 2}  
30 \texttt{STOP}  
40 \texttt{PRINT x + y}  
50 \texttt{FUN}  

\texttt{Results in:}  

\texttt{BREAK IN 30}  

\texttt{hit ENTER for the rest}\ldots \texttt{)}  

\texttt{BREAK in 30}  

\texttt{Now, by typing in \texttt{CONT}, the program will resume execution}  
\texttt{For example:}  

\texttt{CONT}  

\texttt{22}  

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

\texttt{press ENTER to continue}\ldots \texttt{\texttt{)}}  

\texttt{Which do you wish to do}  

A \texttt{\texttt{)}} \texttt{Continue on}  
B \texttt{\texttt{)}} \texttt{Review this section again}  

\texttt{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 the word BASIC, (consult your BASIC manual or your system operator for specific directions on your particular machine) you will be in the IMMEDIATE mode. In this mode, you may execute many BASIC statements without having to type in line numbers or the command RUN. For example:

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

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

press ENTER to continue?

**IMMEDIATE (cont)**

Another example would be:

`PRINT 95+10=49`

Which would result in:

:49;

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.

dress 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.

dress 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.

dress 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 I=2
D STOP

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

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

press ENTER to continue.

What command will erase everything in temporary memory?

Type in the command using capital letters: NEW

CORRECTO MUNDO - THAT'S RIGHT!

press ENTER.

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

Type in the command using capital letters: LIST

RIGHT YOU ARE.
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

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 STAND BY

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).
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 SQR for square root.

Press ENTER.

LIBRARY FUNCTIONS (cont):

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

PRINT SQR(25);

Which would result in:

5

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

Press ENTER.
Listing of Program 'LESSON1/TXT'

LIBRARY FUNCTIONS (cont)

Another example would be:

10 PRINT SQR(2+42)
RUN
which would give you:

9

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

ALL TIME - USE ONLY CAPITAL LETTERS IN YOUR ANSWERS!

Are the library functions stored permanently in memory?

Yes
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
9 Review this lesson again

press the letter opposite your choice and press ENTER? A

Variables

When working with computers, it is necessary to define the type of data you are manipulating, if for no other reason than to communicate your program to someone else. Numbers, such as 10. 12. 20. 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 3, then the computer will store the value 3 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, DPNI) 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 I?!

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 'LESSON17.TXT'**

Which do you want to do?

A. Continue on
B. Review this lesson again

dress the letter opposite your choice and press ENTER: A

**NUMERIC VARIABLES**

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

```
x = 1-2
```

In this case, the value of '1' 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 '1' to 'x', but only temporarily. Take the following example:

10 x = 1-2
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 SEEDING letter OR number to a variable to distinguish it from another. 'AI', 'XY', 'X1', and 'FF' are legal variables. 'IA', 'X1', or 'II' 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 \( j = 7 \)
30 \( i = i + 1 \)

The variable \( i \) is assigned the value of \( i + 1 \) or 11.

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

press ENTER

Numeric variables cont

When you use variables on the right side of an equation, you must have assigned values to the variables previously. It is a "ERROR" 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 \( i = j + 2 \)

In the above statement, \( i \) will be assigned the value of "7", or 7.

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 onto 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 ENTER

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

A 10 \( \text{LET} \ x = 2 \)
B 10 \( \text{LET} \ x = 4 \)
C 10 \( \text{LET} \ x = \text{const} \)
D 10 \( \text{LET} \ x = \text{expr} \)

press the letter opposite the correct answer and press ENTER
A 10 \text{LET} \ x = 2
CORRECT - Good job!
press ENTER

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

A 10 \( A = B \)
B 10 \( C = A-B \)
C 10 \( C = A+B \)
D 10 \( C = A-B \)
**** Listing of Program "LESSON.TYP" ****

press the letter opposite the correct answer

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

press ENTER

which do you want to do?

A Continue on
B Review this lesson again

drop the letter opposite your choice and press ENTER

String Variables

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

There is a special way of defining 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 X 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.

292
***** Listing of Program 'LESSON1.TXT' *****

"C" = "114 West Cottage Street"

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

Examples of illegal STRING labels would be %, %%, $C$, JIM

press ENTER

Which of the following are correct STRINGs?

A. AT$ = "12"
B. AT$ = "12"
C. AT$ = "North State Street"
D. AT$ = "North State Street"

press the letter opposite the correct answer and press ENTER

E. CORRECT - are you sure you are only a student

press ENTER

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

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

"This is the correct answer exactly as it would be printed
" THIS IS A TEST QUESTION"
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:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>addition</td>
<td>A+B</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
<td>A-B</td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
<td>A*B</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
<td>A/B</td>
</tr>
<tr>
<td>^</td>
<td>exponentiation</td>
<td>A^2</td>
</tr>
</tbody>
</table>

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

Press ENTER.

Using Arithmetic (cont.)

Note that a "*" always must be used for multiplication.
If you tried to use an * or a, the computer would think you were trying to put in another variable label. This would give you an error message. Also, you cannot use terms like PI to mean 3.14. If you do, you will get an error.
***** Listing of Program "LETSTL.TXT" *****

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 = (6*2)+5-(2*2-2)  
```

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

Press ENTER.

Using Arithmetic (cont.):

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</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 = 10^2/6-1*2-2

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

Press ENTER

Using Arithmetic

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:

9 = 17 / 3  

is much different than:

9 = 17 / 3

Do you see why? The first value assigned to z is 5.714. The second value assigned is 1.5.  Stud the example carefully.

Press ENTER

Which of the following statements will assign the value of 1.5 to the variable x?
**** Listing of Program "LESS0N1/TXT" ****

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

press the letter opposite the correct answer and press ENTER?

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

press ENTER to continue?

which do you want to do?

A Continue on
B Review this lesson again

press the letter opposite your choice and press ENTER? A

Comparing Variables

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

press ENTER for examples

Comparing Variables (cont)

(symbol table)
We will discuss these in more detail in a later lesson.

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

By now, you should understand the following program:

```
10 PRINT "The product of 10 times 9 is "
20 PRINT 10 * 9
30 PRINT
40 PRINT
50 END
```

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.

LESSON 1B

This is the second part of a two part lesson. It is divided into the following sections.
### LESSON 1B

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

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

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

What is your choice? 7

### FINAL TEST (Lesson 1B)

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

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

Press ENTER to continue.

Is a computer program called Software?

A. Yes
B. No

Press the letter opposite the correct answer and press ENTER.

A. CORRECT

Press ENTER.

Which of the following is an example of a FUNCTION

A. LIST
B. NEW
C. SOR
D. ADD

Press the letter opposite the correct answer and press ENTER.

C. CORRECT

Press ENTER to continue.

Which statement would print the word TEST

A. PRINT TEST
B. PRINT "TEST"
C. OUTPUT "TEST"
D. PRINT 'TEST'
### Listing of Program ‘LESCON1.41’

Press the letter opposite the correct answer and press ENTER.

**CORRECT**

Press ENTER.

What would the following program's output be?

```
10 X = SCR.A
20 PRINT X
RUN
```

Type in your answer and press ENTER.

**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 COMPILER. TRUE or FALSE?

A. TRUE
B. FALSE

Press the letter opposite the correct answer and press ENTER.

**CORRECT**

Press ENTER.
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 in TRUE or type in FALSE for this statement. FALSE

CORRECT

press ENTER

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

A  CPU
B  MEMORY
C  KEYBOARD

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

CORRECT

press ENTER
If you had the following program in memory:

```
10 G=34
20 I=40
30 r=10
40 _=0
```

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

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

Press the letter opposite the correct answer and press ENTER. D

CORRECT

Press ENTER.

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

"YOU HAVE PASSED"

Do you want your score recorded on a permanent file?

A) YES
B) NO

Which? A

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

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

ENTER your word or have now? DAN CREAGAN

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

Press ENTER to return to MENU?

Break in 4180
Ready
SYSTEM*RESET *00*
TRSDOS Ready
BASIC
BASIC 31.30.90 for TRSDOS Version 3
Copyright (c) 1982 by Microsoft, licensed to Tandy Corporation.
All rights reserved.

Ready
run"lesson2

LESSON: BASIC II
VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON: About One Hour

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

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

dress the ENTER i.e. to continue"

LESSON 2

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

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

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.

105
D. I want to return to the Menu.

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

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

Here is one possible way to complete your homework:

10 PRINT "THE SUM OF 2, 1 AND 22 IS"
20 PRINT 2 + 1 + 22
30 PRINT
40 PRINT
50 PRINT
60 PRINT
70 PRINT "THE SUM OF 2, 1 AND 22 IS"
80 PRINT 2 + 1 + 22
90 PRINT

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

Introduction

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

Fileanaes

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

Fileanaes (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)

dress 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.

dress 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 ORGENE1CO (or CPM)

_FILENAME/extension (filename.extension)

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

dress ENTER

Filenames (cont)

TPS-80 ORGENE1CO
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 .<EXTACT>. NUMBERS may be used in both filenames and extensions, but they must NOT be the FIRST letter.

Filename (cont)
Here are some examples of legal filenames for your computer:

MYPROG.BAS
MRHAPPY.BAS
SWIMFIN.BAS
GOODNESS

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.

Filename (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 dir 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) XXXXXX.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) XXXXXX.FIL

press ENTER

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

A  YES
B  NO

Press the letter opposite the correct answer and press ENTER?

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

Press ENTER?

Which do you wish to do?
A  Continue on
B  Review this section again

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

SAVE, LOAD and RUN

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

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

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

Press ENTER?
SAVE, LOAD, and RUN (cont)

There are other things you should be aware of before you try to SAVE a program. First, there should be enough room on the disk to hold the program. If you are using a 32KMEMCO hard disk, you will probably have no problem in this area. Ask your SYSTEM OPERATOR for more information. If you are using a 40-60 small floppy disk, then go to the COMMAND mode by typing in CMD'S and then typing in DIR 1: or DIR A:. Drive 1 is called C: and drive 2 is called D:. Then watch the display. you will see the free space left on the disk, and a DIRECTORY or the files on the disk. If you have over 10 crams, then you have enough room for almost any program. SEE YOUR OPERATING MANUAL FOR MORE DETAILS.

press ENTER

SAVE, UN, and LOAD (cont)

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

SAVE "MYPROG.BAS" or "MYPROG.BAS"
if using a TPB-30

You 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 'LESSON2.TXT' **** 07:11 8/3 - 1985:50

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 FRITZ?

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

Press the letter opposite the correct answer and press ENTER.

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

press ENTER

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

212
A: SQUEEZE
B: LOAD "SQUEEZE"
C: LOAD "SQUEEZE.BAS"

dress the letter opposite the correct answer and press ENTER

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

REMKARKS

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. REMar statements.
dress ENTER

REMKARKS cont

The format for a REM statement is

LINE # REM (REMARKS)

An example is:

114
@@ REM This is how a REM statement is made

when the computer sees REM it ignores all data that follows

press ENTER

Here is an example of REMarks in a program:

C1: R = 2
C2: REM the variable R in the next line is approx equal to PI
C3: R = 3.14
C4: C = R**2
C5: PRINT C
RUN

Which would give:

10.5

Notice that the REM statement was not printed

press ENTER

C6: PRINT "SWEETUMS is a girl.",
C7: REM "SWEETUMS is a girl."
C8: PRINT "her real name is CANDY"
C9: REM "her real name is CANDY"
RUN

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

SWEETUMS is a girl.
her real name is CANDY
"as in 'TRUE or FALSE, whichever is correct"

NOTES - the correct answer is TRUE - REM statements are not printed
**** Listing of Process 'LESSCH11.TXT' ****  
press ENTER

Which do you wish to do?
A  Continue on
B  Review this section again

cress the letter opposite the correct answer and press ENTER

INPUT Statements
we saw in the first lesson that DATA can be assigned to a variable using the equals '=' sign. For example:

10  x = 10
20  PRINT x
RUN

Gives 10:

In this example, we assigned 10 to x on line number 10.
cress 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

C19
*** Listing of Program "LESSON2.TAP" ***

press ENTER

**INPUT Statements (cont)**

1) INPUT"press the correct letter, then press ENTER"%R

Gives just:

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 "A", just as if we had assigned a value to it in an equals statement. Also note that a semi-colon is placed after the text:

press ENTER

**INPUT Statements (cont)**

Here is another example:

10 A = 1
20 INPUT"Enter a number between 1 and 2.\n20 \n30 B = A
40 \nGives just:

Enter a number between 1 and 2
if we ENTER a 2 then

press ENTER

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

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

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

Press ENTER

INPUT Statements

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

[1]: PRINT "When you see a question mark, ENTER a 5"
   2: INPUT 
   3: PRINT "When you see a question mark, ENTER a 5"
      4: INPUT

Press ENTER

[1]: PRINT "When you see a question mark, ENTER a 5"
   2: INPUT 
   3: PRINT "When you see a question mark, ENTER a 5"
      4: INPUT

Press ENTER
Notice now the INPUT statement droops (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 a number between 1 and 9
FIN

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

```
10 INPUT ENTER three I. numbers?A.B.C
```

```
ENTER three I. numbers? 10
" A
" B
" C
```

Press ENTER

INPUT Statements (cont.)

```
10 INPUT ENTER three I. numbers?A.B.C
```

```
ENTER three I. numbers? 10
" D
" E
```

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

Press ENTER

INPUT Statements (cont.)

```
10 INPUT ENTER three I. numbers?A.B.C
```

```
ENTER three I. numbers? 10
" F
```

```
***** Listing of Program "LESSON1.TXT" *****

ENTER three (3) numbers: 10.12.16

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

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

Press ENTER?

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

ENTER the word "TRUE" or ENTER the word "FALSE"

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

Press ENTER?

What is the prompt that an INPUT statement ALWAYS gives?

A) A question mark
B) No question marks
C) Quotes
D) The word "INPUT"

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

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

Press ENTER?

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

1) INPUT A,B,C

2)
**** Listing of Program "LESSON.16" ****

A T.TEN.15
B 1.1.7.15 North Elm
C "ANON.15"
D "NONE.D"

ENTER the letter opposite the correct response and press ENTER.

WRONG - the correct answer is B (1.1.7.15 North Elm)
press ENTER

which do you wish to do?
A Continue on
B Review this section again
press the letter opposite the correct answer and press ENTER? A

GOING TO SECOND PART - PLEASE STAND-BY

LESSON 2B

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

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

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

333
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 way. 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.

clear ENTER

READ and DATA (cont)

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

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

clear ENTER

READ and DATA (cont)

The format of the READ statement is:

'line number': READ 'variable or variables:'
An example of a READ statement that would read values into variables Bl, A, and X$ is:

10 READ Bl,A,X$
press ENTER

READ and DATA (cont)

10 READ Bl,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 Bl, A, and X$ would hold 22, 15, and "AIN'T she sweet?" respectively.

press ENTER

READ and DATA (cont)

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

press ENTER

Is the following statement TRUE or FALSE?
**Listing of Program 'LESSON2.TAT'**

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,11$  
20 DATA 21.5  "AIN'T she sweet"'

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

press ENTER

**READ and DATA (cont.)**

10 READ B1,A  
20 PRINT B1,A  
20 PRINT "The average of these numbers is":  
30 PRINT B1 A+B1/2  
30 DATA 21.10.15  
50 PRINT

Gives us:

5 10 15  
The average of these numbers is 10

WOW! Study this example for a moment. Note that the calculations were printed beside the message instead of below it.
**** Listing of Program 'LESSON/TAT' ****

press ENTER?

READ and DATA (cont):

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

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

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

press ENTER?

READ and DATA (cont):

Let's look at it once more:

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

Gave us:

5 10 15
The average of these numbers is 10

press ENTER?
**** Listing of Program 'LESSON2.TIT' ****

Now for a neat example:

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

Gives us:

OUT OF DATA IN 10

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

Press ENTER?

READ and DATA (cont)

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

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

Press ENTER?

READ and DATA (cont)

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

Gives us:

1 2
Although there were more data points that could have been read, there were no more variables left to read, so the program stopped. This would not cause an ERROR message.

press ENTER

Is the following program valid?

10 READ X,Y,Z
20 DATA 25.2,15.55.64
30 PRINT Z

a. Yes
b. No

Enter the letter opposite the correct answer?

WRONG - the program is valid

press ENTER

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

WRONG - the correct answer is 15. Z is the third variable to be read, so the third data point is 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,B1
20 PRINT A1,B1
30 RESTORE
40 READ C1,D1
50 PRINT C1,D1
or DATA 1,2
RUN

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)?
**** Listing of Program 'LESSON2/TI' ****  

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

Press ENTER.

Which do you want to do?
A: Continue on
B: Review this lesson again

Press the letter opposite your choice and press ENTER. A

Branching Introduction

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

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

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

Branching Introduction

10 A = 0
20 A = A+1
30 IF A = 5 THEN GOTO 50
40 GOTO 20
50 PRINT A
60 END
Line 30 is CONDITIONAL because it will only GOTO line 50 if the CONDITION that \( A = 5 \) is satisfied. That is, control will only be transferred to line 50 if \( A = 5 \).

Press ENTER.

Branching Introduction

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

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

Press ENTER.

Branching Introduction

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

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

Study this carefully, and then press ENTER.
Branching Introduction

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

The statement in line 20 has made a COUNTER out of the variable A. 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 = 5 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: 1
B: 2
C: No output would come from this program
D: 3

Press the letter opposite the correct answer and press ENTER.

WRONG, the correct answer is D (3)

Press ENTER.

Which do you want to do?
A: Continue on
B: Review this lesson again

Press the letter opposite your choice and press ENTER: A

**IF Statements**

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

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

Press ENTER.

**IF Statements (cont)**

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

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

If we ENTER a 12 when prompted by line 10, the CONDITIONAL statement in line 20 will not be satisfied. Nothing will happen until control passes to line 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
The IF statement can also be used to compare two expressions such as:

```plaintext
IF \( j + e \) THE4 1OTO 19i)
```

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

```plaintext
IF \( 0t2-3) \leq 3 \times 10 \) THEN A=1
```

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

```plaintext
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.

```plaintext
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.

```plaintext
IF Statement (cont)
```

```plaintext
\( \)\( \)
```
IF (true/false expression) THEN (action) ELSE (action)

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

press ENTER

What is the output of the following program?

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

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

END

ENTER the letter opposite the correct answer:

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

press ENTER

The IF statement is a CONDITIONAL statement.

Is the above sentence TRUE or FALSE

A) TRUE
B) FALSE

choose the letter opposite the correct answer and press ENTER
WRONG - the IF statement IS a CONDITIONAL statement.

press ENTER?

Which do you want to do?

A Continue on
3 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?
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 10"
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?

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

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

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

Press ENTER.

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

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

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

Press the letter opposite the correct answer then press ENTER?

WRONG - the correct answer is A "30 and 40"

Press ENTER.

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

Remember, after you are done here, practice some of the
things you have learned. And keep a BASIC manual by your side. This program will show you the fundamentals. You have to teach yourself how to be good at BASIC. That means you must PRACTICE. 

press ENTER to go to the MENU.

LESSON 2B

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

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

4) I'm taking this part in its entirety.
3) I wish to review selected areas (or 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. B

LESSON 2B

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

1) READ, DATA and RESTORE 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 9 and press ENTER to return to the Menu.

what is your choice? 5
FINAL TEST (lesson 2)

His 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

Which of the following is a legal filename?

A SCOTIE.CMB (in CPM or Cromemco)
B "TRIUMPH/650" (in TRS-80)
C THEWAYOF.B21 (in CPM or Cromemco)
D SCORE6.DAT (in TRS-80)

Enter the letter opposite the correct answer:

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

Press ENTER

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

A CREATE "LOVE"
B RUN "LOVE"
C LOAD "LOVE.BAS"?
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. 

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

A. TRUE
B. FALSE

Enter the letter opposite the correct answer.

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

Press ENTER.

Which of the following examples is INVALID?

A. INPUT T
B. INPUT"ENTER YOUR NAME":IN
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?

343
A SAVE
B RUN
C LOAD
D STORE

ENTER the letter opposite the correct answer

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

press ENTER

Which of the following statements is legal?

A READ A B C1
B DATA A B C
C READ 22.33,44
D DATA "QUIT","FIRE"

ENTER the letter opposite the correct answer

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

press ENTER

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

A TRUE
B FALSE

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

press ENTER?

The IF statement is a CONDITIONAL BRANCHING statement.
A TRUE
B FALSE

ENTER the letter opposite the correct answer?

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

press ENTER?

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

10 N = 5
20 ON N GOTO 40,50,20
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/T1' ****

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

press ENTER
***** Listing of Program "LESSON2.TXT" *****

YOU HAVE NOT RECEIVED ENOUGH POINTS TO PASS

YOU SHOULD RETAKE LESSON 2!

you will be returned to the Menu.

press ENTER to return to the MENU?

Break in TDDO
Ready
SYSTEM"RESEXE" TDD
TRSDOS Ready
BASIC
BASIC 01.10.00 for TRSDOS Version 3
Copyright (c) 1982 by Microsoft. Licensed to Tandy Corporation.
All rights reserved.

Ready
RUN 'LESSON1'

LESSON: BASIC 3
VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON: Less than one hour

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

OBJECTIVE: "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 & WJLL 4) FOR NEXT Statements
2) LOOPS (Intro) 5) Advanced FOR NEXT
3) COUNTER variables

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.
I want to return to the menu.
Press either capital A, B, C, or D and then press ENTER.

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 discussed 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 make the mistake of killing someone else's files, or your own by accident. Remember, KILL does just what it says, it KILLS files.

Press ENTER to learn about the KILL command.

Introduction & KILL (cont.)

The format for the KILL command is:

KILL "filename.ext"

You must enclose the filename in quotation marks.
The extension is only needed when the original file had one.

Once this command is entered, the file will be removed unless protected by passwords. You can learn about passwords from your system manual. If the file is needed, 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:INT" ****

Which of the following statements is legal?

A) Kill STUPID
B) Kill "STUPID:100"
C) Kill "STUPID.BAS"

Press the letter opposite the correct answer and press ENTER.  C

Correct

Press ENTER.

Which do you wish to do?

A) Continue on
B) Review this section again

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

LOOPS (Intro.)

One of the most exciting aspects of computer programs is their ability to 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 8, the middle column would be the squares of the figures in the adjacent first column...
and the third column would be the square of the figure in the
adjacent middle column. It would look like this:

| 1 | 1 | 1 |
| 4 | 16 | 256 |
| 9 | 81 | 916 |
| 16 | 256 | 4096 |

Press ENTER.

If you were to write separate print statements for these
calculations, you would have to write at least 3 lines of code.
Using Loops, you can calculate this column of figures, and
aclelelt 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. C1 = 1
2. IF C1 = 0

Press ENTER.
**** Listing of Program "LEGENDARY" ****

10 IF I = 10 THEN STOP
20 GOTO 10

The value of I after running this program is 10. Press ENTER.

COUNTERS (cont.)

10 I=0
20 I=I+1
30 IF I = 10 THEN STOP
40 GOTO 20

The COUNTER variable is I. It is INCREMENTED 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. Press ENTER.

If you see what would happen if we didn't include the IF statement? The program would look like this:

10 I=0
20 I=I+1
30 GOTO 20

The program would never stop. It would be caught in an ENDLESS LOOP. We have many terms for this common mistake. Some examples: LOCKED UP, KNEE, and *** DUMMY, you did it again: ***

Press ENTER.

END
COUNTERs (cont)

If your program ever gets hung up, you may have to type 
"CONTROL C" or CCR or Creadsol, or (BREAK) iif 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 Creadsol 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 loaded 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 I = 0
20 X = 0
30 R = 1
40 IF I = 2 THEN I = 2
50 IF X = 5 THEN STOP
60 PRINT I
70 GOTO 10
80 END

4 I is the COUNTER
B X is the COUNTER
C BOOTHARE COUNTERS, but it will control the program
press the letter opposite the correct answer and press ENTER B
CORRECT - that shows good understanding of the principle
press ENTER
which do you wish to do?

A. Continue on
B. Review this section again

press the letter opposite the correct answer and press ENTER A

FOR - NEXT Statements

In the previous discussion of counters, we showed you that we tested a counter with an IF statement to see if it had reached a desired level. If it had, then we went to another part of the program, or STOPped. There is a set of statements in BASIC that lets us shorten the statements needed to duplicate this kind of program. The set of statements is called FOR - NEXT.

press ENTER

FOR - NEXT (cont):

<table>
<thead>
<tr>
<th>COUNTER loop</th>
<th>FOR NEXT loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 REM</td>
<td>10 FOR X = 1 to 5</td>
</tr>
<tr>
<td>20 X = X + 1</td>
<td>20 PRINT X</td>
</tr>
<tr>
<td>30 PRINT X</td>
<td>30 NEXT X</td>
</tr>
<tr>
<td>40 IF X = 5 GOT0 90</td>
<td>40 END</td>
</tr>
<tr>
<td>50 GOT0 20</td>
<td></td>
</tr>
<tr>
<td>20 END</td>
<td></td>
</tr>
</tbody>
</table>

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

COUNTER loop

FOR NEXT loop

10 IF x = 0
20 PRINT x
30 NEXT x
40 GOTO 10
50 GOTO 70
60 END

Line 10 of the FOR NEXT loop initializes x, the same way that lines 10 and 20 of the COUNTER loop do. The FOR NEXT loop sees line 10 as 'I am going to start a loop. X is the counter. I will begin with x = 1 and when x is GREATER than 0 I will GOTO the statement that is after the NEXT x statement'.

press ENTER

COUNTER loop

FOR NEXT loop

10 IF x = 0
20 PRINT x
30 NEXT x
40 GOTO 10
50 GOTO 70
60 END

Line 10 of the COUNTER loop and line 20 of the FOR NEXT loop are the same and perform the same function. Note that the PRINT statement was indented a few spaces in the FOR NEXT loop. This lets you see the loop structure better. You should do that in your own programming, too.

press ENTER

COUNTER loop

FOR NEXT loop

50
10 LET X = 0
20 LET X = X + 1
30 PRINT X
40 IF X = 3 THEN GOTO 60
50 GOTO 20
60 END

The IF statement in line 40 of the COUNTER loop determines if X has reached 3. This was done automatically by the FOR NEXT loop because the last value in line 10 was specified as 3.

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. It is an example, any VARIABLE works. The NEXT statement causes an UNCONDITIONAL GOTO to the FOR statement. The variable is incremented and the FOR word tests to see if it EXCEEDS value #2. If it does, then control passes to the statement that follows the NEXT statement.

If there are no statements passed the NEXT word, the program ENDS.

press ENTER

FOR NEXT (cont)

This concludes the initial FOR NEXT section. You should understand what a simple FOR NEXT statement does. If you do not, you will be given a chance to review this section before you go on. But before that - QUIZ time.

press ENTER
Which of the following programs made the above list?

A 10 FOR X=1 TO 3
   20 PRINT X*2
   30 NEXT X

B 10 X=1
   20 PRINT X*2
   30 IF X=3 THEN GOTO 10
   40 GOTO 10

C 10 FOR I = 1 TO 4
   20 PRINT I*2
   30 NEXT I

D 10 PRINT I*2
   20 PRINT I*2

Press the letter opposite the correct choice and press ENTER? A

CORRECT:

press ENTER?

The NEXT statement is the same as an UNCONDITIONAL BRANCHING Statement. It branches to the next line under the FOR statement.

Is the paragraph above TRUE or FALSE?

A TRUE
B FALSE

ENTER the letter opposite the correct answer" A

WRONG - the NEXT statement is UNCONDITIONAL, but it branches to the SAME line as the FOR statement. It also increments the variable before the FOR statement tests it.

Press ENTER"
Which do you wish to do?

A Continue on
B Review this section again

Press the letter opposite the correct answer and press ENTER: A

Advanced FOR NEXT

10 FOR X = 1 TO 10 STEP 2
20 PRINT X
30 NEXT X

Gives us:

```
1
3
5
7
9
```

Interesting, isn't it? .... press ENTER for an explanation?

Advanced FOR NEXT

10 FOR X = 1 TO 10 STEP 2
20 PRINT X
30 NEXT X

The STEP word in the FOR statement caused the value of X to be incremented by 2 instead of 1. The output of 1,3,5,7,9 was correct because the NEXT statement is the statement that actually increments the value, so the first time through, 1 was equal to 1; the next time through it was equal to 3; etc.

Press ENTER
Advanced FOR NEXT

10 FOR X = 10 to 1 STEP -2
20 PRINT X
30 NEXT X

Gives us:

10
9
8
7
6
5
4
3
2

press ENTER for explanation?

10 FOR X = 10 to 1 STEP -2
20 PRINT X
30 NEXT X

In this case, we STEpped DOWN instead of up. Notice that the first value of the FOR statement is the largest value. It would not make sense to specify a negative STEP and give the range of a positive STEP. For example:

10 FOR X = 1 to 10 STEP -1

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

press ENTER

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

10 FOR X = 1 TO 2
20 FOR Y = 1 TO 2
30 PRINT X, Y
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)

```plaintext
10 FOR i = 1 TO 2
20 FOR y = 1 TO 2
30 PRINT i, y
40 NEXT Y
50 NEXT I
```

Try to figure how the program produced the columns above it and then press ENTER.

On the first pass through the program, the FOR NEXT loop for 
I was set up, and control passed to the lines between FOR i = 
1 to 2, and NEXT I. Those lines happened to be another loop 
with the variable Y as the FOR NEXT variable. When the i FOR 
NEXT executed the first time, i was equal to 1 and Y was 
printed twice, once as a 1 and once as a 2. When the i loop
finished, the NEXT I statement caused control to GOTO line 10 again. The process was repeated, but X now was equal to 2.

press ENTER?

```
1          1
1          2
2          1
2          1

10 FOR I = 1 TO 2
20      FOR Y = 1 TO 2
30      PRINT I,Y
40      NEXT Y
50      NEXT X
```

Can you figure it out now? Think about it and then press ENTER?

```
10 FOR I = 1 TO 2
20      FOR Y = 1 TO 2
30      PRINT I,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 I statement.

press ENTER?

The format for nesting loops is:

```
LOOP 1 (for)
   LOOP 2 (for)
      LOOP 3 (for)
```

**** Listing of Program 'LESIOJ/TXT' ****

LOOP 3 (next)
LOOP 2 (next)
LOOP 1 (next)

Notice that you must back out of a nest in reverse order.

press ENTER?

Is the following program valid?

10 FOR X = 1 to 200
20 FOR Z = 1 to 2
30 PRINT X+2
40 NEXT Z
50 NEXT X

A YES it is a good program
B NO it is not a valid program

ENTER the letter opposite the correct answer?

WRONG - the correct answer is A. There is nothing wrong
with the program.

press ENTER?

Give the first statement of a FOR NEXT loop, that will
cause the loop to increment the variable I from one to 20 in
steps of 2. Use 10 as your line number and leave one space
between all words, numbers, and variables.

ENTER your answer?

WRONG - the answer is:

10 FOR I = 1 TO 20 STEP 2

press ENTER
Is the following program valid?

10 FOR I = 1 TO 3
20 FOR J = 1 TO 3
30 PRINT "This is the last question"
40 NEXT I
50 NEXT J

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

ENTER the letter opposite the correct answer?

A wrong - 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 36
3 9 36 81
4 16 64 256
5 25 125 625
6 36 256 1296
7 49 1225 14401
8 64 4096

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

It looks like child's play now, doesn't it"
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)  2) Arrays
2) Arrays  3) Dimension & Clear  4) Test

Please select your answer:

4  I'm taking this part in its entirety.
5  I wish to review selected areas. I'm taking the test.
6  I want to go to the first part.
7  I want to return to the Menu.

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

Arrays (Intro)

We have seen that variables are storage places for data. In
large programs, it is difficult to manipulate large data bases
without having numerous variables to assign the bits of data
to. However, there is a way to group our variables into
similar bunches that makes it easier for us to tell what part
of the data base our variable belongs to. We can use
**Arrays (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.

**Name** | **Rank** | **Training Status**
--- | --- | ---
John Doe | Foreman | 0
Jake Rodinski | Peon | 3
Mark Muffin | Specialist | 5

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)

**Name** | **Rank** | **Training Status**
--- | --- | ---
John Doe | Foreman | 9
Jake Rodinski | Peon | 3
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.

```plaintext
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 them ARRAYS so that we could hold nine bits of data.
As we continue through our lessons, we will discover some very powerful uses for ARRAYS.
Is the following ARRAY and its subscript valid?

A(I)

A TRUE
B FALSE

ENTER the letter opposite the correct answer: A

Wrong - the correct answer is A

press ENTER?

Which do you wish to do?

A Continue on
B Review this section again

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

ARRAYs

If you have a little mathematics in your background, you will have noticed that ARRAYs are almost the same as their math equivalent, except that the subscripts are in parenthesis instead of slightly lower and to the right of the variable.

The previous examples all dealt with a ONE-DIMENSIONAL ARRAY. That is, there was only one number in parenthesis that was significant. ARRAYs with TWO, THREE, FOUR, or more dimensions are possible. Most dialects of BASIC, including Microsoft, will handle at least 3 dimensions. An example of a TWO dimension ARRAY would be X(I,J). Notice that the extra dimension was designated by just adding another subscript in-
side the parenthesis. A THREE DIMENSIONED ARRAY looks like:
this: A(2,1,9) or T(5,44,3) (or any combination of number)
press ENTER?

10 FOR X = 1 TO 2
20 FOR Y = 1 TO 2
30 READ A(X,Y)
40 PRINT A(X,Y):
50 NEXT Y
60 NEXT X
70 DATA 5,10,15,20

The above program combines several of the techniques that we
have been learning. Before we tell you the answer, try to
figure out what the output of the program would be. We warn
you, it is a little tricky, but see if you can figure it out.

press ENTER?

10 FOR X = 1 TO 2
20 FOR Y = 1 TO 2
30 READ A(X,Y)
40 PRINT A(X,Y):
50 NEXT Y
60 NEXT X
70 DATA 5,10,15,20

5 TO 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
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.

criss 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.

criss ENTER

Y
1. 2.
1----1----
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?

4 x=1, y=1
5 x=1, y=2
6 x=2, y=1
7 x=2, y=2

ENTER the letter opposite the correct answer

WRONG - the correct answer is C

press ENTER
2. 1:5 1:20 |
[-----[-----]

If X = 1 and Y = 1, what value is in the pocket pointed to by
ARRAY A(X,Y)?

A 10
B 15
C 20
D 5

ENTER the letter opposite the correct answer?

WRONG - the correct answer is D

press ENTER?

1. 2.
[-----[-----]
1. [ S I 10 ]
[-----[-----]  ARRAY A(X,Y)
2. [ 15 I 20 ]
[-----[-----]

The FIRST subscript in an ARRAY ALWAYS indicates the ROWS of
DATA, and the SECOND subscript indicates the COLUMNS of data.
Therefore, ARRAY A(X,Y) has X rows and Y columns. You will
find that once you visualize a two dimensional ARRAY, the
others will come quite easily.

press ENTER?

which do you wish to do?

4. Continue on
9. Review this section again

press 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 pockets that you will use. Up to a limit, the machine can handle unexpected ARRAYS, but after you designate more than 10 pockets or 3 dimensions, the computer must know ahead of time so it can reserve enough space. The way you tell the computer to use an ARRAY with at least one subscript bigger than 10 or with more than 3 dimensions to it (as WITH ORIENT), 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):11
80 NEXT x
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 and Clear

Another problem the computer has with memory allocation is reserving enough room for STRING space. For every letter in a STRING, the BASIC processor must use a little over one memory pocket.
List... of P'scram

LESSON

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 memory locations, so it must always be the first statement if you are going to need it.

press ENTER?

Dimension and Clear

If you use the CLEAR statement in the middle of a program, the accumulation of data that you have stored in variables to the point that the CLEAR word was used, will be zeroed out.

10 CLEAR
20 CLEAR
30 PRINT X
40 RUN

Do you see? The CLEAR word zeroed out X".....press ENTER"

10 CLEAR 1000
20 DIM G(50), A$(1000)
30 FOR X = 1 TO 100
40 READ A$(X)
50 NEXT

The above shows the first 4 lines of a program that is going to use more than 1000 spaces of string space (that will give an average of 10 letters per pocket of A$(X) and is going to use 50 pockets of the numeric array G. (note that you always get pocket 0 for free. it is normally not used)
Notice the position of the CLEAR and DIM statements. Press ENTER.

What will be the output of the following program?

10 A(I,J) = 5
20 CLEAR
30 PRINT A(I,J)

A 0 because the CLEAR statement is in the wrong place.
B 5 - the CLEAR statement only affects STRINGS.
C Nothing because A(I,J) will cause an ERROR

Enter the letter opposite the correct answer: 

Wrong - C is the right answer

Press ENTER.

The CLEAR word is used to clear storage space for strings, but it also wipes out other data.

A TRUE
B FALSE

Enter the letter opposite the correct answer: A

Correct: Goodyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy
the computer will not automatically reserve more space than needed. For example:

10 DIM A(1,2);

This would be a valid, and memory conserving statement. The processor wouldn't reserve a 10 X 10 socket ARRAY for A(1,1). It would only reserve a 2 X 2.

What is wrong with this program?

10 CLEAR 1000
20 DIM A(1)
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

cpress 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 questions). Use only capital letters in your answers. Don't include extra spaces or letters. GOOD LUCK

Press ENTER to continue?

Which of the following is valid:

A KILL "SLEAZY"
B KILL SLEAZY
C UNSAVE "SLEAZY"
D UNSAVE SLEAZY

ENTER the letter opposite the correct answer? A

CORRECT

Press ENTER.

Does the FOR NEXT combination have to be used to set up a loop?

A YES
B NO

ENTER the letter opposite the correct answer? B

CORRECT

Press ENTER.
Write out the first line of a FOR NEXT loop using \texttt{I} as the variable, start the loop at 1 and end it at 5. use a STEP of 2. use Line number 50. Use all caps. leave one space between all terms.

ENTER your answer

WRONG - the correct answer is 50 FOR I = 1 TO 5 STEP 2
See part 1. FOR - NEXT.

press ENTER\

10 FOR \texttt{X} = 1 TO 5 STEP 2
20 PRINT \texttt{X};
30 NEXT \texttt{X}
RUN

ENTER the output from this program. leave one space between terms. \textit{hint: be sure to consider the semi-colon in line 20)}

ENTER your answer\texttt{^1^3^5}

CORRECT

press ENTER\

How many sections (or elements) can an array have without a \texttt{DIM} statement?

A 10 not counting the 0 element
B 11 not counting the 0 element
C 10 not counting the 0 element
D 4 not counting the 0 element

ENTER the letter opposite the correct answer\texttt{^A}
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(12,2,2)
Enter YES or NO? YES
CORRECT
press ENTER?

Which of the following is valid
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.

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. LOOPS
part 2. CLEAR statement

press ENTER?
Do you want your score recorded on a permanent file?
A YES
B NO
Which? B

You are now qualified to go to LESSON 4.

If you want a homework assignment, select it now.

Do you want to see your homework (Y/N)?

Break in 3720
Ready
SYSTEM*RESET*00
Listing of Program 'LESSON4/TXT' ****

TRSDOS Ready

BASIC

BASIC 31.00.00 for TRSDOS Version 6

Copyright 1983 by Microsoft, licensed to Tandy Corporation.

All rights reserved.

Ready

RUN 'LESSON4

LESSON:  BASIC 4       VERSION:  1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON:  About 1.5 hours

AUTHOR:  Capt. Danny J. Creagan

Air Force Institute of Technology

OBJECTIVE:  To teach the student how to make the computer

communicate with standard personal devices.

Press the ENTER key to continue?

LESSON 4

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

into the following sections:

1) Introduction         3) Sequential Files Intro.
2) OPENING \\ LIST        4) OPENing Files

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

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

Introduction

In this lesson we will start learning how to communicate with our disk drives and printers. The first section deals with printing our programs on a printer. We can either print the output (ie the answer that our program calculated), or we can print our program listing to a printer.

The remaining sections will show us how to store data on a disk, so that we can save important calculations for future use.

Both sections are very important to the programmer and you will find yourself using them often.

Press ENTER.

Introduction

From this lesson to the end of your training, we will be covering areas that are complex and difficult to remember. Therefore, we recommend that you have your BASIC manual with you at all times, when we ask you a question, and you are not sure about the answer, LOOK IT UP IN YOUR MANUAL. The answer will also be in the lesson, but you should get used to using the manual. You cannot memorize all the rules in a few weeks or months. So be sure to keep your reference book handy, whether you are taking a test, or taking your own program.

Remember, looking up the answers is not cheating. It's LEARNING.

Press ENTER.
**** Listing of Program 'LESSON4.TEX' ****

**PRINT & LLIST**

**PRINT** 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 Coloraco, 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.

**PRINT & LLIST**

To **PRINT** a STRING to the printer, you must enclose it in quotes, just like the PRINT statement. When you **PRINT** 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 **PRINT** statements are:

```
**PRINT** "MONTHLY TRAINING REPORT"
```

Press ENTER.

**PRINT & LLIST**

You can **PRINT** THES also, just as you can PRINT THES, however, the THF function reacts differently on different machines. The general format for **PRINT**ing a THF list:

```
**PRINT** "THF 1", data
```

Press ENTER.
where $L$ is a number between 1 and the length of your printer carriage. When used, the carriage will go over $L$ 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.

cress ENTER

LPRINT & LLIST

LLIST works just like LIST only it outputs to the printer.

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

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

cress ENTER

LPRINT and LLIST output data to the printer and to the screen.

Is the above sentence TRUE or FALSE?

A TRUE
B FALSE

ENTER the letter opposite the correct answer and press ENTER.

CORRECT

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

A) LIST
B) LIST ALL
C) LIST "ALL"
D) LIST "fn" (fn is your program name)

ENTER the letter opposite the correct answer: A

CORRECT

press ENTER

which do you wish to do?

A) Continue on
B) Review this section again

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

Sequential Files Intro

Remember, in the previous lessons, we learned that we could INPUT data into our program, 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 this lesson, we will learn how to store our data that we ENTERED on a disk FILE. When we do that, we can always recall it for future use, and we won't have to keep entering the same information every time we run a program. We just have to tell the program to read the data from a disk. Our lives will be 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.

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 most of the commands are similar. Your BASIC manual will explain the differences. DIRECT access is often called RANDOM access by some manuals.

Sequential Files

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. They would be stored in a line, and the computer would have to start with the first age and search through the list in order until it found the last age. Even though this is not the best way of handling files, the computer still does a good job of it. Your scores for the previous tests were logged into a file using SEQUENTIAL access mode.

Press ENTER.

If you wrote two names to a disk file using SEQUENTIAL access mode, could you get to the 9th name you wrote without reading the first name you wrote?

A: YES
B: NO

I wish to take the 5th

ENTER the letter opposite the correct answer: A

WRONG - the correct answer is B

Press ENTER.

Sequential Files Intro

The statements and functions used with sequential files are:

[Further text not visible in the image]
***** Listing of Program "LESSON4.TST" *****

OPEN  PRINT#  SRF
CLOSE  INPUT#  CLOSE

We will be covering these words in greater detail in the following sections. You don't have to memorize them now. Just look them over. Generally, you would OPEN the file, either INPUT* your data, or PRINT* it out to the disk, and then you would CLOSE the file before going on.

press ENTER

1: OPEN "C:\\1\"TEST"\n2: INPUT* Use your name and press ENTER*17\n3: PRINT*17\n4: CLOSE ;

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 1 opens the file (we'll explain more later). Line 2 asks the operator to INPUT his/her name, and line 3 writes the name out to a file called "TEST". Line 4 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 for 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 "somewhere in-between"

ENTER the letter opposite the correct answer" 2
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 are SEQUENTIAL and DIRECT.

Is the above statement True or False?

A. True
B. False

Enter the letter opposite the correct answer - A

Correct!

Press ENTER

Which do you wish to do?

A. Continue on
B. Review this section again

Press the letter opposite the correct answer and press Enter - A

OPEN statement

10 OPEN "D:\DATA\TEST"
10 INPUT "Type in your name and press ENTER":
10 PRINT "":
40 CLOSE

In this ATE program, line 10 opens the file we wish to save. Whenever you work with a file, you MUST OPEN it first, then separate 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 "LESSON4.TAT" ****

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

press ENTER

Can you OPEN a file that is already OPEN?

A  YES
B  NO

ENTER the letter opposite the correct answer A

CORRECT

press ENTER

10 OPEN "D:\TEST"
10 INPUT "be in your name and press ENTER"
10 PRINT
10 CLOSE

Line 10 OPENs 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 up to 8 buffers more on some versions. For our
purposes, we will use buffer #1. The last bit of data is the
filename. Notice that the file mode and filename are in quotes

press ENTER

10 OPEN "D:\TEST"
10 INPUT "be in your name and press ENTER"
10 PRINT
10 CLOSE
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-
80. Just answer the BASIC startup dialog with the correct
number of files - 3 is the default!. The filename must be in
quotes.

press ENTER

What are the two file modes?

A  INPUT and OUTPUT
B  "I" and "O"
C  1 through 4 and 4 through 8
D  None of the above

ENTER the letter opposite the correct answer

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 "D:U.M.TEST"
2) INPUT "Type in 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'4" *****

A  ONE  B  TWO  C THREE  D  FOUR

ENTER the letter opposite the correct answer. A

CORRECT
press ENTER

1. OPEN "FILE1","TEST"
2) INPUT"Use your name and press ENTER" AS
3) PRINTFILE.A$ 
4. CLOSE 1

You must enclose the filename in quotes if it is a character string. However, the following is also legal:

10 AS $ = "TEST"
20 OPEN "C".1,.AS

press ENTER

10 AS $ = "TEST"
20 OPEN "C".1,.AS

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 FILE$" RESULTS
***** Listing of Program 'LESSON4/TXT' *****

A  YES
B  NO

ENTER the letter opposite the correct answer: B

CORRECT
press ENTER?

10 OPEN "D:". "TEST"
20 INPUT "Type in your name and press ENTER": T
30 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 does NOT 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
30 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 file name that was entered in line 20!

Sequential files must be loaded into memory, manipulated and
then written back out in their entirety. You cannot just write
a single record onto the front of the file that's on the disk.
Write in the statement that will OPEN a file for OUTPUT to disk and use buffer number J. The filename is TWIG.

Do not use a line number (although you would normally), and do NOT put in ANY blanks.

ENTER your answer?

*ERROR - 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"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 'Q'
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?

*ERROR - the correct answer is B

press ENTER?

Which do you wish to do?

A. Continue on
B. Review this section again
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.
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

CLOSE Statement

we already stated previously, that the CLOSE statement was necessary after you were done manipulating your files. It has a few variations that are nice to know.

CLOSE (buffer #, . . . )

The format for CLOSE is the BASIC word CLOSE plus an optional buffer number. If you include a buffer number, just that buffer will be closed. If you do not include a buffer number then ALL the buffers that were OPEN will be closed.

press ENTER

is the following program valid (lines 20 & 30 are good):
***** Listing of Program 'LESSON4/TIT' ***** 07/11/82 - 02:13:01

10 OPEN"C:\J\QUESTION"
20 INPUT A$
30 PRINT A$
40 CLOSE
50 OPEN"C:\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 41 was CLOSED by line 40. so buffer 3 could be used again in line 50.

press ENTER?

Which do you wish to do?

A: Continue on
B: Review this section again

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

PRINT 0

10 OPEN "0:J:\TEST"
20 INPUT"Type in your name and press ENTER":T$
30 PRINT T$
40 CLOSE

10 line 20 we PRINT to the file buffer (that we OPENed in line
10) the variable T$. Notice that the number to the right of the PRINT$ corresponds to the buffer number. If the buffer
was number 3, then, after we OPENed the file, we would
PRINT$3, T$ in line 20

press ENTER
PRINT \n
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).

press ENTER\n
PRINT\n
Commas and semicolons react the same way with PRINT\$ that they do with PRINT statements. If you were to write the following program:

10 A=10.1
20 B=20.2

and save it as an RT, then you would put this on the disk:

10.1 20.2

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

press ENTER\n
PRINT\$\n
'he above command puts this on disk:

10.1 20.2

If you use a semicolon, like this, PRINT\$ then you get:

10.1 20.2
There are only three spaces between the numbers. So to save space on the disk, you may want to write to disk using semicolons instead of commas between your variables. Either way will work, it's just that the commas cause 10 extra blanks to be PRINT'd to the disk.

press ENTER

What is wrong with the following program?

10 OPEN"*."."NEWREC"
10 PRINT#1."PRIVATE"
10 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 PRINTing.

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 & EDF

INPUT is similar to the INPUT word that we learned earlier
but it INPUTS data from a disk that previously had data

400
**** Listing of Program 'LESSON4.TXT': ****

printed to it.

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

The format for the INPUT# statement is similar to the PRINT# statement. It is: INPUT# (buffer#,(variable1),(var2),(etc)

press ENTER

INPUT#

INPUT# inputs data from a sequential disk file and stores the data in a variable. INPUT# doesn't care how data was placed on
the disk. It could have been put there with one PRINT#
or twenty PRINT# statements. WHAT MATTERS TO INPUT# IS HOW THE
DATA IS TERMINATED ON THE DISK, AND WHAT KIND OF DATA IT IS
INPUTTING.

press ENTER

Does the INPUT# statement check to see how the data was
placed on the disk. or does it check to see how the data is
terminated?

A It only checks to see how it was terminated - it doesn't
care how the data got there.

B It checks to see how the data was placed, It makes a
difference how many PRINT# statements were used.

C How THIS question is easy. Just give me a second and I'll
think of the answer. Hmnn, let's see...No, don't tell me...

ENTER the letter opposite the correct answer. A

CORRECT - good job
INPUT$ & EDF

If we are inputting STRING data (our variable is a STRING such as INPUT$.), INPUT$ 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 EDF marker (more about EDF 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 EDF marker.

Press ENTER

INPUT$ and EDF

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 COMPILER 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 RETURN AND COMMAS.

Why is this important? Because you wrote the data to disk with "40 VARIABLES, AND YOU READ IT BACK WITH ONLY ONE. IF YOU TRIED TO READ THE NUMBER AFTER YOU READ 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."
20 PRINT#1,1001

And when you read it back, use two separate statements.

10 INPUT#1.A$  
20 INPUT#1,N

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

press ENTER

10 OPEN "C:\1\TEST"
20 INPUT#1,N  
20 PRINT #1   
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  
20 PRINT #1   
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 it puts a marker on the end of it. The marker is called an EOF marker. If we wanted to find out if the end of a file had been reached, or if the file was empty, we would test it like this:

```
10 OPEN"*I:*.TEST"
20 IF EOF(1) THEN GOTO 30
30 INPUT11, N1
40 PRINT N1
50 GOTO 30
60 PRINT "END OF FILE ENCOUNTERED"
70 CLOSE 1
.............more of the program or an END statement
```

```
10 OPEN"*I:*.TEST"
20 IF EOF(1) THEN GOTO 30
30 INPUT11, N1
40 PRINT N1
50 GOTO 30
60 PRINT "END OF FILE ENCOUNTERED"
70 CLOSE 1
.............more of the program or an END statement
```

If we had printed a thousand and one names into the file ‘TEST’ the computer would put the EOF marker in place # 1002 and when we used the EOF buffer # test after name # 1001. control
would pass to line 50. 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 #: 

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 111 "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 "...TEST"
2) IF EOF THEN STOP
3) INPUT #1, #
4) PRINT #
5) SOTO #1

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.

cress 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.

INIT: the program would have to read the data into an array and then write 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.

cress ENTER

10 CLEAR 500 ; clears string space
20 INPUT "Input filename":$%3
30 INPUT "Output filename":$%3
40 OPEN 1,...,%3 ; opens the INPUT file
50 OPEN 2,...,%3 ; opens the OUTPUT file
60 IF EOF 1 goto 111 ? checks for end of file on file #1
70 INPUT T1,...,%3 ; inputs data from file #1
80 PRINT T1,...,%3 ; prints data to the new file
90 GOTO 40 ; goes back for more from file #1
100 INPUT 'ENTER DATA enter 999 to stop':$%3
110 IF $3 = 999 goto 140
120 PRINT T2,...,%3 ; updates new file with new data
130 GOTO 30 ; goes back until line 111 sees 999
140 CLOSE

we assume a legal input file previously existed...press ENTER
which do you wish to do?

A Continue on
B Review this section again

Press the letter opposite the correct answer and press ENTER

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

Going to TEST 4 - please stand by

FINAL TEST (Lesson 4)

This test consists of 10 questions. You must get 70 percent
of them correct to pass. (That's 7 right out of the 10 ques-
tions!). Use only capital letters in your answers. Don't
include extra spaces or letters. When you successfully complete
the test, you will be given a homework assignment that will
bring everything together for you. GOOD LUCK!

Press ENTER to continue
When you LPRINT data you must be sure the printer is on, has enough paper, and is properly connected. LPRINTing does not send output to the screen.

Is the above paragraph TRUE or FALSE?

A  TRUE
B  FALSE

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER

What is the command to list your program to the printer?

A  LPRINT "(filename:"
B  LIST "(filename:"
C  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

WARNING - the correct answer is B

We did not access a record in the middle of the
file is to read the records in front of it. 
See part 1: Sequential Files.

press ENTER

To transfer data from your file to a disk, the computer must
first send the data to a buffer where it is processed. The same
is true for transferring data from the disk back to the
computer.

Is the above paragraph TRUE or FALSE?
A TRUE
B FALSE

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

If you wish to access a previously created disk file, you will
read in the data and use it to make an important financial
decision. Which of the following statements will open the file
and read the data into buffer E. The filename is B52DAT

A 1. OPEN*",1,*B52DAT*
B 2. OPEN*"5",*B52DAT*
C 3. OPEN*",5,*B52DAT*
D 4. OPEN*",5,*B52DAT*
E None of the above

ENTER the letter opposite the correct answer? E
CORRECT
press ENTER
What is wrong with the program below (assume the END statement in line 20 automatically closes the files when it is executed):

1) OPEN "FILE.TXT"
20 IF EOF: THEN END
30 INPUT #1, A$
40 PRINT A$
50 GOTO 10

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

ENTER the letter opposite the correct answer: B

CORRECT

press ENTER

The CLOSE word can be used without a number; however, when you do that, only the most recent file is CLOSED.

Is the above statement TRUE or FALSE?

A: TRUE
B: FALSE

ENTER the letter opposite the correct answer: A

CORRECT

press ENTER

What is wrong with the following program?

1) OPEN #1, "TEST"
20 PRINT #1, "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 #2 and read the data into variable #3.

ENTER the command now: INPUT#2,#3
CORRECT
press ENTER

What is wrong with the following program:

10 IF EOF THEN GOTO 40
20 INPUT#1,#4
30 GOTO 10
40 CLOSE
50 END

A Nothing
B The file wasn't properly opened
C Nothing will happen with the data
D The END statement is not needed

ENTER the letter opposite the correct answer: B
***** Listing of Program "LESSON4/EX" *****

CORRECT

does 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

does ENTER

Do you want your score recorded on a permanent file?

A = YES
B = NO

 Which?

You are now qualified to go to LESSON 5.

Do you wish a homework assignment this week?

Proceed from here.

READ SYSTEM "LESS" 400
TRSDOS Ready

LESSON  BASIC 5  VERSION: 1 AUGUST 87

TIME REQUIRED TO COMPLETE LESSON: About one hour

AUTHOR:  Capt. Joseph J. Crescenzi
Air Force Institute of Technology

OBJECTIVE:  To teach the student how to use SUBROUTINES and LIBRARY functions.

Type the ENTREP 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 SPOOL
5. GO TO RETURN

--- Select this part of the lesson...---
1. I want to review selected areas.
2. I want to go to the second part.
3. I want to return to the start.
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. We are on the
downhill part of the course. That is right. We 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 that not have to memorize
them. We'll just have to know how they work, and then go on
to Lesson 6.

press ENTER

INTRODUCTION

Somewhere during each lesson, we emphasize the value of doing
extra study. Besides what this course teaches you, we will learn
more when we DO something that we have read about. This is
particularly true with learning a programming language.

For now, though, set out your favorite BASIC manual and turn
us beside your computer for another lesson in BASIC.

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
SUBROUTINES

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

Typically, we use subroutines to do tasks which we perform often in our program such as printing out a menu after each module or if the program is finished by a student.

Then we, the programmers, can write the menu program once and call on it when we need it.

Press ENTER

Would we want to use a subroutine?

1. Do those parts of the program that are used often
2. Do inside the normal routines

ENTER: the letter indicates the correct answer

DONE - this was supposed to be an easy question to answer.

Indeed, we screwed up completely. We will send you back to the reading or this part. "Reading between the lines a little.

Press ENTER

SUBROUTINES

We use the word "ROUTINE" to describe the statements in the code of a program. "SUBROUTINE" is used to describe a subroutine 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 mini-programs that we put in our main program and when we need them, we can go to them, perform the task they are built to do, and return to the main program.

Subroutines differ from other forms of program control in that they always return control to the statement that follows the statement that called them. That means you can call a subroutine anywhere within a program, and the computer will remember where the call came from, and when the subroutine has done its job, control will go back to the statement that followed the calling statement.

Press ENTER
SUBROUTINES

If you have built a program that has to continually print out a prompt, asking the user 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 DAI program is a computer assisted instruction"
2. print "A program that teaches students."
3. SHOW 10:10 "1" LDB use sub. after ever. major part
4. IF '1' = "G" THEN 500 10:10 '1' returns from sub with choice 5. ..... ELSE
6. SHOW 10:10 "1" LDB use sub. after ever. major part
7. END: If sub. ensures program doesn't go to sub by accident
8. END
9. "The subroutine follows"
10. END
11. "INPUT Do you want to continue or review what you've just done?"
12. END

We will ensure 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
Enter the letter opposite the correct answer: 


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

**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 SOSUB the main program from a subroutine without risking disaster. You should always use the RETURN statement.

press ENTER

**print** "A CAI program is a computer assisted instruction"

**print** "Instruction that teaches students."

1. **SOSUB (X):** (X) ---- LOAD use sub. after every major part
2. IF T = '1' THEN GOTO 11: (X) TA returns from sub with choice
3. ......ETC
4. **SOSUB (X):** (X) ---- LOAD use sub. after every major part
5. **END** (X) You must ensure program doesn't goto sub by accident
6. **REM**
7. **REM** the subroutine follows
8. **REM**
***** Listing of Program "LESSON7.12" *****

1000 INPUT "Do you want to continue or review #2 Con B=Rev. #1?"
1010 RETURN

Line 24 and line 31 call the subroutine, and line 10410 RETURNS to the appropriate statement.  press ENTER

10 print "A QBI program is a computer assisted instruction"
20 print "What program that teaches students."
30 GOSUB 10000 " use sub. after even major part
40 IF 1=6 THEN GOTO 10 : 1% returns from sub with choice
50 ....STC
60 GOSUB 10000 " use sub. after every major part
70 END  "You must ensure program doesn't goto sub by accident
80 REM
90 REM the subroutine follows
100 REM
1100 INPUT "Do you want to continue or review #2 Con B=Rev. #1?"
1110 RETURN

If line 70 calls the subroutine, what line gets control after the RETURN statement .... ENTER the correct line number" 5:

CORRECT = 5

press ENTER

GOSUB 4 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
**** Listing of Program 'LESSON2.CPP' ****

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 past 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 100 and put the space between all terms.

Enter the command now: 100 GOSUB 1100

CORRECT

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: A
NESTED SUBROUTINES

what do you think the output of the following is?

10 PRINT "Main Program"
20 GOSUB 100
30 GOSUB 200
40 END
50 PRINT "Subroutine One"
60 GOSUB 200
70 RETURN
80 PRINT "Subroutine Two"
90 RETURN

press ENTER for the answer:

The answer is:
10 PRINT "Main Program"
20 GOSUB 100
30 GOSUB 200
40 END
50 PRINT "Subroutine One"
60 GOSUB 200
70 RETURN
80 PRINT "Subroutine Two"
90 RETURN

Main Program
Subroutine One
Subroutine Two
Subroutine Two

press 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
40 RETURN
50 GOSUB 100
60 RETURN

"We put the line numbers as they would be executed, putting a space between each number. i.e. 10 20 30 401
Enter your answer?"

ANSW - the correct answer is: 110 120 20 29: 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
Remember the ON GOTO statement from our previous lessons?

There's a command very similar to ON GOTO that can be used with subroutines. You have probably already guessed that the command is called ON GOSUB.

Here is an example:

```plaintext
10 INPUT"ENTER A NUMBER BETWEEN 1 AND 4";
20 ON N GOSUB 200,400,500,600

If N = 1 then subroutine 200 would be executed, if N = 2, subroutine 400 would execute, and so on.

PRESS ENTER
```

ON GOSUB

```plaintext
10 INPUT"ENTER A NUMBER BETWEEN 1 AND 4";
20 ON N GOSUB 200,400,500,600

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 200)

PRESS ENTER FOR AN EXAMPLE
```

```plaintext
10 PRINT"TYPE 1 FOR SQUARE TABLE"
20 PRINT"TYPE 2 FOR THE CUBE TABLE"
30 INPUT A
40 ON A GOSUB 1000,2000
50 GOTO 10
1000 FOR I = 1 TO 50
1100 PRINT F.I
1200 NEXT I
1300 RETURN
1000 FOR I = 1 TO 50
1100 PRINT F.I, I*I
1200 NEXT I
```

424
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

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

You are now done with this part of the lesson. If you ENTER a 'C' you will go on to the second part. ENTER an 'R' to start over. Which do you want (C or R)? C

LESSON 5A

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

1. Introduction 4) User Defined Functions
2. Functions Overview 5) DEF Statements
3. Library Functions 6) TEST

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

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

In this lesson we will tackle the different 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*

FUNCTIONS OVERVIEW

Library functions contain useful operations that have been
written and stored in the computer, and are there whenever you need them. User functions are functions that you make up by inserting the instruction to make them in your program.

Then, when your program needs the special USER FUNCTION, it can call on it with a special word.

There are many functions, and, depending on the specific implementation of Microsoft BASIC, you probably have at least 12 library functions stored in your computer. On the next screen is a list of the typical set of library functions.

press ENTER

Typical Library Functions

1: ABS(exp) - gives absolute value 7: LOG(exp) - gives LOG(e)
2: ATN(exp) - arctangent in radians 8: RAND() - random numbers
3: COS(exp) - returns cosine of exp 9: SGN(exp) - sign of exp
4: EXP(exp) - natural exponential 10: SINT(exp) - sine of exp
5: FIX(exp) - gives integer of exp 11: SQRT(exp) - square root
6: INT(exp) - gives integer of exp 12: TAN(exp) - tangent(exp)

exp = an appropriate numeric expression - remember, you can't use ** EXP** for some of the functions (SQRT(-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 may be used at any time in your program.

A: TRUE
B: FALSE

ENTER the letter opposite the correct answer: A
CORRECT
press ENTER

Is the following sentence TRUE or FALSE?

User Functions are made by the user and are not normally stored in the computer.

A TRUE
B FALSE

ENTER the letter opposite the correct answer? A

CORRECT
press ENTER

Which do you wish to do?

A Continue on
B Review this section again

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

Library Functions
1) ABS(exp) - gives absolute value 7) LOG(exp) - gives LOG(exp)
2) ATN(exp) - arctangent in radians 8) RND(0) - random numbers
3) COS(exp) - returns cosine of exp 9) SGN(exp) - sign of exp
4) EXP(exp) - natural exponential 10) SIN(exp) - sine of exp
5) FIX(exp) - gives integer of exp 11) 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 ANY number for some of the functions (SGR(-1) bombs!)

Here are twelve of the most used library functions. As an
example of how much time they can save. think of the number of
statements you would have to write to calculate the logarithm
of a number.
press ENTER

Library Functions

All you have to do to calculate the log of a number, say the
number is 10, is to type in the statement - PRINT LOG(10).
The example of the logarithm of 10 is exactly how we implement
the library functions. Here is another example:

10 INPUT"Type in the number you want the square root of":N
20 PRINT 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"Type in the number you want the square root of":N
2: PRINT SQRT(N):
3: GOTO 10

Notice that the function allows you to use a variable as an
argument for the number you want the square root of. In
addition to this, you can use a FUNCTION as an argument for a
FUNCTION.
press ENTER for an example?

Library Functions

429
10 INPUT "Enter the number you want the square root of": N
20 PRINT SQR(ABS(N));
30 GOTO 10

In this example, no matter what the sign of the number you enter, you will get a valid square root, because ABS(N) will give the absolute value of N as a positive number, and then SQR(ABS(N)); will give the square root.

You can use just about any valid numeric expression for the first 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 SQR(ABS(N));
30 GOTO 10

The expression ABS(N); is evaluated first, then the outside expression - SQR(---); - is evaluated next. Remember from the first lesson when we said that the parenthesis is the highest priority arithmetic expression? That means that any expression that is within parenthesis will be evaluated first. If more
than one set of parenthesis is used, then the expression with-
in the innermost set of parenthesis is evaluated first.

press ENTER.

Is the following sentence TRUE or FALSE?

You can write your own library functions in special cases.

A  TRUE
B  FALSE

ENTER the letter opposite the correct answer! A

WRONG - Library functions are permanently stored in the
computer and cannot be created. The correct answer is B

press ENTER.

In the next few screens, we will ask you questions concerning
library functions. You should get out your BASIC manual and
look up the answer to the questions before you attempt to enter
your response.

You should give your answer in the format, FUNCTION(number).
You will always be given the number for the function, and you
do not have to include a line number. For example, if we ask
for the function that gives the square root of 10, you would
type in: SQRT(10).

press ENTER.

What is the function that gives the natural log of e? LOG(2)

CORRECT

press ENTER.
Name the function that gives the absolute value of -3? $\text{ABS}(-3)$

CORRECT
press ENTER.

What is the function that gives the arctangent of $\pi$?

WRONG - the correct answer is $\text{ATN}(-\pi)$
press ENTER.

What is the function that gives the sine of $0.5$? $\text{SIN}(0.5)$

CORRECT
press ENTER.

Name the function to give a random number between 0 & 1? $\text{RND}$

WRONG - the correct answer is $\text{RND}(10)$
press ENTER.

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 FN12 = INT(RAND*6) + 1)
20 DEF FMM(A, B) = A * B/2 + (A - B)
30 X = FN12
40 PRINT X
50 Y = FMM(1, 4)
60 PRINT Y
70 END
There are two user functions in this program. They are defined in lines 10 and 20, and called on in lines 30 and 50. We will cover this program in greater detail in the next section.

Press ENTER.

Which do you wish to do?

4 Continue on
5 Review this section again

Press the letter opposite the correct answer and press ENTER.

DEF Statement

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

(Line #) DEF FN func. name(=vars) = (expr)

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

Press ENTER.

1: DEF FN1 = INT RAND(0) * 10
2: DEF FN2 = A * B * C - A + 5
3: i = FN2
4: PRINT i
5: i = FN2 A
6: PRINT
TO END

Jet out your BASIC manual and look up DEF. It will show you examples similar to this. Line 10 defines a function named
R2 that will be set equal to the expression on the right side
of the statement. It will return a random number between 1
and 10. Note that this user function uses a library function
as part of its definition. This is legal.

Press ENTER.

Can you use a library function as part of the definition of a
USER function?

A: YES
B: NO

ENTER the letter opposite the correct answer: A

CORRECT

Press ENTER.

10 DEF FNR2 = INT(RAND(10) + 1)
20 DEF FWMAB = A * B + (A - B);
30 X = FNR2
40 PRINT X
50 X = FWMAB
60 PRINT X
70 END

Line 70 calls on the function defined in line 10. When it
uses, 1 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 state-
ment. It sends X, a random number; and the number 4. A
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 DEFINE 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?

The two values that are defined as A and B will be set equal to the corresponding values of X and Y. A will equal X and B will equal Y. The DEF statement will then use these numbers to calculate the (exop) 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/TXT' ****
***** Listing of Program 'LESSONS.7AT' *****

40 PRINT x
50 y = FNW.x, A
60 PRINT y
70 END

What value would be passed to B in the second DEF statement?

A 4
B The value of A
C Cannot tell - not enough information

ENTER the letter opposite the correct answer: A

CORRECT

Press ENTER

10 DEF FNW-A = LEN(A)
20 A$ = "This is a function"
30 x = FNW(A$)
40 PRINT x

Here is an example of passing a string in a user function.
In all the examples that we have shown, we have always told
the computer when we wanted to invoke a user function by using
the pre-"FN"var". That is the only way to get to your
user function.

Press ENTER

10 DEF FNW-A = LEN(A$)
20 A$ = "This is a function"
30 x = FNW(A$)
40 PRINT x

Note in the example above, that we passed a string within a
numeric variable, but that in the DEF statement, we set the
numeric variable equal to LEN(string). LEN(string) is an
example of yet another type of FUNCTION that we will study in
the next lesson. LEN(string) returns a numeric value equal to
the LENGTH of the string that is within the parenthesis.
Don't worry about it now. We will cover it in the next lesson.
However, because it is a NUMERIC value, its type matches the
DEF statement, and therefore it is valid.

Press ENTER.

What kind of function do you create with the DEF statement?

A USER
B LIBRARY
C EITHER USER or LIBRARY
D DEFINED FUNCTIONS

ENTER the letter opposite the correct answer? A

CORRECT - but that was an easy one. Try the next question if
you think you are up to it.
We know you are - we're just trying
to lighten things up a bit.

Press ENTER.

Is the following statement valid?

10 DEF FN88.A.B = A + B

A No. to call on it simply but FN88 somewhere in your program
B No, the terms A and B are not defined and will cause errors
C No. 88 is not a valid variable for this position
D No, the syntax is good, but the beat is poor, you can't
catch to it.

ENTER the letter opposite the correct answer?

WRONG - the correct answer is D

Press ENTER.
Which do you wish to do?

A  Continue on
B  Review this section again

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

You have completed this lesson and now you can go to the test. If you wish to review parts of this lesson, type in an 'R'. If you wish to continue to the test, type in a 'C'.

Enter your choice (R or C)? C

Going to test number 5 - wait patiently please.

FINAL TEST (lesson 5)

This test consists of 10 questions. You must get 70 percent of them correct to pass. (That's 7 right out of the 10 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.

Type 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 1000 20 30
CORRECT
press ENTER?

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

10 INPUT A
20 ON A GOSUB 1000,2000,3000
30 END
... rest of program is not important

A Nothing
B Subroutine 3000 would be called
C Subroutine 1000 would be called
D The program would end

ENTER the letter opposite the correct answer: D
CORRECT
press ENTER?

how many RETURNS can you have in a subroutine?

A One
B As many as you want, but they should be kept to a minimum
C One for every GOSUB
D No more than the amount of memory available

ENTER the letter opposite the correct answer: B
CORRECT

446
What is wrong with this program?

10 INPUT A
20 GOSUB 1000
30 END
4000 IF A = 10 THEN RETURN
5000 IF A = 11 THEN RETURN
6010 GOSUB 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
A 5
B 525
C the natural logarithm of 25
D 100

ENTER the letter opposite the correct answer: A

CORRECT
press ENTER

What type of functions would be written by you
Type in your answer using all capital letters. Do NOT append the word FUNCTION on the end of your answer.

Type in your answer now: USER.

CORRECT
press ENTER

What is the term you would use to define a USER function.
Hint: It's two words. DON'T include a variable, leave a space between the two words.

ENTER your answer now: DEF FM

CORRECT
press ENTER

10 DEF FMX = 10 * X
20 Y = FMX

What is the term that would complete line 20 if we wished to call on the user function in line 10?
***** Listing of Program 'LESSON7/7' *****

ENTER your answer now? FMX
CORRECT
press ENTER?

Which of the following functions is valid?

A) DEF FN(L,M) = 2 * M/2 + A$  
B) DEF FN1(L,M) = 2 * C + 2 * M  
C) DEF FN2(A,B) = LEN(A$) + 1  
D) DEF FN3 = SQR(-4)  
E) DEF FN4 = 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've 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 1820
Read
SPLIT*RESET *SD
LESSON: BASIC 6

VERSION: 1 AUGUST 83

TIME REQUIRED TO COMPLETE LESSON: Less than one hour

AUTHOR: Capt. Larry D. Criden
Air Force Institute of Technology

OBJECTIVE: To teach the student about string functions
and the Microsoft Editor.

Press the ENTER key to continue.

LESSON 6

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

1: Introduction
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.
INTRODUCTION

This is your last lesson! Congratulations! If you have taken the previous five lessons, you should be feeling a little more comfortable with Microsoft BASIC by now.

In this lesson we will cover STRING functions first, and then we will learn about the Microsoft Editor (in the second half).

As we learned before, a BASIC string is one or more alphanumeric characters that are treated as a single collection of data. Using the concepts in this chapter, you can perform many of the same types of operations on STRINGS that you can perform on numeric data.

Press ENTER.

INTRODUCTION

As a small review, you should remember that string data can be designated in two ways. You can assign your data to a string variable, or you can enclose the data in quotes. Here are two examples:

10 PRINT "This is one way to designate a string"
20 V$ = "Another way is to put it in a variable and print it"
30 PRINT V$

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

Press ENTER.

INTRODUCTION

10 PRINT "This is one way to designate a string"
20 V$ = "And this is another"
30 PRINT V$
40 PRINT V$
This is one way to designate a string

And this is another

Here is another variation of our little program. Note that
"And this is another" is not printed until line 40 is executed.

press ENTER

INTRODUCTION

Remember, in this lesson, as in all of our lessons, you should
have either a good BASIC manual handy, or you should have
an experienced programmer around to help you with difficult
problems.

In some of the answers you will need to be sure you use the
correct case (either uppercase or lowercase), so be sure to
read all the questions carefully.

Set 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 $S$ is blank)

10 READ A$
20 INPUT A$
30 LET A$ = S$
40 A$ = "NOW IS THE TIME"
50 DATA "NOW IS THE TIME"

A. No, the LET statement in line 30 is illegal
B. No, the string assignment in line 40 is illegal
C. No, you cannot read data into a string (line 10 is bad)
D. Yes, all statements are legal

ENTER the letter opposite the correct answer?

WRONG - all these assignments are legal

cress ENTER?

String Assignment

10 READ A$.S$
20 PRINT A$
30 A$ = S$
40 PRINT A$
50 DATA "THE GRINCH", "IS COMING"
RUN

THE GRINCH
IS COMING

Notice that A$ was converted to S$, and all the data was printed out by using just A$ in print statements.

cress ENTER
String Assignment

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

Notice the dollar sign is always included with a string variable. The dollar sign tells the computer to treat the variable as a string instead of as a numeric. Also, whenever you assign data to a string, it must either be another string or it must be enclosed in quotes.

Press ENTER.

Assign NOW IS THE TIME to a string variable called NN$ and use line number 10 as your statement number.

Put one blank between terms.

ENTER your answer now?

WRONG - the correct answer is 10 NN$ = "NOW IS THE TIME"
(You could have also answered 10 NN$ = "NOW IS THE TIME")

Press ENTER.

You can also INPUT string data from an external file.
(The following program assumes that a file named TEST was previously created on disk)

10 OPEN"*.", "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

dress 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
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$, BS
20 PRINT A$
30 A$ = BS
40 PRINT A$
50 DATA "THE GRINCH", "IS COMING"
RUN

"THE GRINCH
IS COMING"

Notice that A$ was converted to BS, and all the data was
printed out by using just A$ in PRINT statements.
press ENTER

String Assignment

10 READ A$, BS
20 PRINT A$
30 A$ = BS
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 \texttt{NOW IS THE TIME} to a string variable called \texttt{NN$}
and use line number 10 as your statement number.

Put one blank between terms.

\texttt{ENTER your answer now ?}

\texttt{WRONG} - the correct answer is \texttt{10 \texttt{NN$ = 'NOW IS THE TIME'}}
\texttt{(you could have also answered \texttt{10 \texttt{NN$ = 'NOW IS THE TIME'}} )}

\texttt{press ENTER?}

You can also \texttt{INPUT} string data from an external file.
the following program assumes that a file named \texttt{TEST} was
previously created on disk:

\texttt{10 OPEN "...TEST"}
\texttt{20 IF EOF(1) THEN END}
\texttt{30 INPUT A$}
\texttt{40 PRINT A$}
\texttt{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 \texttt{BASIC} idea is the same.

\texttt{press ENTER?}

\texttt{which do you wish to do?}

\texttt{H} \texttt{Continue on}
\texttt{E} \texttt{Review this section again}

\texttt{press the letter opposite the correct answer and press ENTER? A}
String ARRAYS

You can assign string data to arrays in the same way as you assign numeric data to arrays. Nearly all the rules are the same. The following is an example.

10 CLEAR 2000
20 DIM A$(100)
30 x = 0
40 x = x+1
50 INPUT "ENTER up to 99 strings. ENTER 'END' to stop":A$(x)
60 IF A$(x) = "END" THEN END
70 GOTO 40

press ENTER

10 CLEAR 2000
20 DIM A$(100)
30 x = 0
40 x = x+1
50 INPUT "ENTER up to 99 strings. ENTER 'END' to stop":A$(x)
60 IF A$(x) = "END" THEN END
70 GOTO 40

Notice the CLEAR statement. Remember that you normally have only 50-100 characters of string space available, and if you are going to need more, you need to tell the computer. Also, note the DIM statement - we need declare our array size if it is over 10.

press ENTER

10 CLEAR 2000
20 DIM A$(100)
30 x = 0
40 x = x+1

465
***** Listing of Program 'LESSON01.TXT' *****

50 INPUT "ENTER up to 99 strings. ENTER 'END' to stop":A$(1)
50 IF A$(1) = "END" THEN END
70 GOTO 40

The variable X acts 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 zeroize all number variables
D To clear the screen
E To help the programmer understand more clearly

ENTER the letter opposite the correct answer: E

CORRECT.

cress ENTER?

String ARRAYS

10 CLEAR 2000
20 DIM A$(1902)
20 X = 0
40 X = X+1
50 INPUT "ENTER up to 99 strings. ENTER 'END' to stop":A$(1):
50 IF A$(1) = "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 a string array?
If you don't use the DIM statement ... type: "answer now"

454
***** Listing of Program 'LESSON7.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 dimension array A$(X)?

4. PRINT A$(5)
0. PRINT A$
5. READ A$(X)

ENTER the letter opposite the correct answer?

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

Concatenation
you may link two strings together by using the 'plus' symbol.
For example:
1) A$ = 'where'
2) B$ = 'Book'
3) C$ = B$ + A$
4) PRINT C$: you could have said 'PRINT B$ + A$' too.
RUN
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$:
40 PRINT FLASH + DANCE:
50 PRINT DANCE + FLASH:
```

Enter the letter opposite the correct answer: C

Correct
Press ENTER.

Which do you wish to do?

- Continue on
- Review this section again

Press the letter opposite the correct answer and press ENTER: A

String Functions

For this section you will definitely need your BASIC manual, so get it out now.
As with arithmetic functions, there are STRING functions. STRING functions are used to manipulate or explore the contents of a string. On the next screen there are several examples of STRING functions. We will go over several of these, but you will not have to memorize them. Rather, you should understand that if you need to access or modify any kind of string, you can probably find a string function that will do the job for you. String functions can be used as part of USER functions as you saw in lesson 5.

Press ENTER for some examples of string functions.

String Functions

1) ASC(string)
2) CHR$(code)
3) FREL(string)
4) INHE:
5) LEN(string)
6) MID$(string, position, length)
7) RIGHT$(string, length)
8) LEFT$(string, length)

At first glance, these functions look like a lot of GREEK. In fact, they look pretty bad at second glance! However, they really are pretty easy to use once you understand them. The best way to learn how to use them is to make a short program and use them the at a time until you see what they do. We will do over examples of a couple to help you watch 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
**** Listing of Program 'LESION46.TAP' ****

Press ENTER

String Functions

10 A$ = "Machine rings"
20 PRINT ASC(A$)

ASC string is a function that returns the ASCII code of the first character of the string. ASCII stands for 'AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE.' Look up the ASCII code for the first letter of A$ in your BASIC manual. What is it? You should have found it to be 73 decimal. When your computer writes data files to disk, it usually writes them in ASCII code, one letter at a time. This function has use when you are trying to convert characters to their number equivalent.

Press ENTER

String Functions

PRINT CHR$ 73;

CHR$ e.g. returns the opposite of the ASC string function. It returns a character equivalent of decimal 73. Which is M.

Press ENTER

What is the output of the following program?

10 PRINT ASC "A"
20 PRINT ASC $"ab"
***** Listing of Program 'LESSONS/TAT' *****

C = 3
D = E
C

ENTER the letter opposite the correct answer:  \[ \]
CORRECT = GREAT!
press ENTER

String Functions

10 \# = "My aching fingers"
20 PRINT LEN(\#)

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("TOM SWIFT")?

A: 11
B: 8
C: 1
D: 0

ENTER the letter opposite the correct answer:  \[ \]

What is the correct answer is \[ \]
press ENTER

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

1) \# = "My aching fingers"
2) PRINT LEFT\#.ASH.2)

- M.
- M.
- M.
- M.
- M.
- M. aching

459
***** Listing of Program "LESSON171" *****

1: Nothing will be output

ENTER the letter opposite the correct answer".

CORRECT

press ENTER.

10 AS = "My aching fingers"
20 PRINT MID$(AS,4,1)

ENTER the output of this program aching

CORRECT - I'm glad to see you use the book!

press ENTER.

10 AS = "My aching fingers"
20 PRINT RIGHT$(AS,4)

ENTER the output of this program.

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

460
**** Listing of Program 'LESSONS/TXT' ****

will 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

cress the letter opposite the correct answer and press ENTER? a

which do you wish to do?
A Continue on
B Review this section again

cress 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
**** Listing of Program 'LEIS0UN/TAT' ****

2: Starting (EDIT/exit) 5: Exit
3: nSPACEBAR. n(L)search 6: n(L)change, n(insert)
7: TEST

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

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 delete a whole BASIC line just because of one
blunder. Using the editor makes it very easy to alter the line.

Until you get familiar with the editor, you may wish to take a
little "cheat sheet" so you can have the commands available on
your visual screen so you can follow along.

Press ENTER.

INTRODUCTION

The object of this part 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 Micro-
soft Manual.

Press ENTER.

482
Starting

The editor is line oriented, meaning that you operate on one line at a time (and not on a screen of data like you do with a word processor).

Throughout this lesson, we will be using one example line to illustrate all the commands. That line is listed below.

\[
10 \text{ FOR } I = \text{EDIT} \text{ PRINT } I \text{ NEXT } I
\]

The columns form what is called a MULTI STATEMENT line. Each time a colon is entered, the computer treats the data following it as a new line. Therefore, on the above line we have three statements. Obviously, there are several errors in the line. Press ENTER

Starting

\[
10 \text{ FOR } I = \text{EDIT} \text{ PRINT } I \text{ NEXT } I
\]

If we wanted to EDIT this line, we would type in the word EDIT, followed by the line number. In this case, we would type in EDIT 10. You can enter the EDITor in other ways, but for this lesson we will always use the EDIT (line number) syntax. Always access the editor from the IMMEDIATE mode.

If you want to EXIT the EDITor after you are done editing, then you just press ENTER. Pressing ENTER from the EDITOR mode updates the line, and puts you back in IMMEDIATE mode again.

Press ENTER

Starting

\[
10 \text{ FOR } I = \text{EDIT} \text{ PRINT } I \text{ NEXT } I
\]

EDIT 10

10
In the above example, we typed in the word EDIT 10, and the
computer put us in the EDIT mode. Note that the line number
we are EDITing appeared. The cursor would normally be flashing
just to the right of the line number.

press ENTER

Type in the command to edit line 10

ENTER the command now? EDIT 10

GOOD .... now
you are in the EDIT mode. What key do you press to exit the
EDIT mode? ... Press the key now ... DON'T PRESS 'BREAK' or
'CONTROL' C'

EDIT 10
10 FOR X = SQL677 : 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 to the right of the line number. So aheas 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 = SQL677 : PRINT = : NEXT 2
EDIT 10
10 FOR X = SQL677 : PRINT = : NEXT 2

4=4
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.


Notice the lowercase "i" in the title to this section. That
refers to a feature of the editor that allows you to position
the cursor 'i' spaces to the right. That is, you won't have
to pound away at your spacebar to get to the 20th character
of a long line. All you do is press the number of characters
you want to "spacever" over and then press the spacebar.
In this example, you are in the edit mode and you want to go
over 10 spaces.

Press a number, up to 300, and then press the spacebar.

L FOR I = 0 TO 300 PRINT I: NEXT I
EDIT ()
L FOR X

"40H. That was tough. If you want to do it again, ENTER a 'Y'
else ENTER an 'N' ".


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

Also, if you were actually using the editor, you would stay
in the EDIT mode, with the cursor over the nth character, after
you pressed the spacebar.

or, we wanted 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 'search' feature.
***** Listing of Program 'LESSON/TEST' *****

Press ENTER

<SPACEBAR, nSEARCH> 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 back-splits 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. Now you do is press 'B' and then press I. The editor recognizes upper and lower case, so be sure the case is right. Do sheed and do it now. You may search for any letter, just type $ first, then type the letter

" FOR $ = s-to $ PRINT $; NEXT $
" FOR $ = s-to $ PRINT $; NEXT

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

<SPACEBAR, nSEARCH>

You will have noticed that the cursor stopped BEFORE the character that you were searching for. That is what is supposed to happen. In addition, if you asked to search for a non-existent character, the computer printed the whole line.

No doubt you will also have noticed that there is a lower case 'n' in the title nSEARCH. It simply means that you can search for the nth occurrence of the specific character. For example, if we were in the EDIT mode for line 10, and we wanted to find the second occurrence of the letter 'B' then we must type '/BB' and the cursor would also over to the 5th column.

Press ENTER

403
Here is an example of the search feature.

```
10 FOR t = set177: PRINT " NEXT 1"
20 EDIT (1)
30 "Now type ISF to find 2nd occurrence of R or 10"
40 IF R = set17 10
50 "The cursor would stop just before the nth occurrence of R."
60 press ENTER
```

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

ENTER your answer now: ISF

CORRECT = OPEN

correct

Which do you wish to do?

- Continue or
- Review this section again

correct the letter corrisponds the correct answer and press ENTER

"C" exits.
***** Listing of Program "LESSON 14" *****

-We are running in to the way the tape and paper at
Microsoft are doing things, and we have too much trouble
with this command. It does what it looks like it does. It
deletes characters, one at a time, or "n" at a time.

For example, if you are in the EDIT mode for line 1 and you
want to delete the NEXT character, then just press 0 and the
character will be excised 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:

100 FOR K = 0:PRINT NEXT

*EDIT*

100: Now say, we want to delete the "e" character.
    All we do is put the cursor to the left
    of the two characters and press 0 twice

1000 FOR N = 1 TO 2:PRINT NEXT: NEXT

Then it would look like this:

Then we would press ENTER and we would have this

But, try the example and read the appropriate paragraph in the

press ENTER

"Delete"

100 FOR N = 0:PRINT NEXT

Now it is your turn. You have to EDIT the above line so that
the characters "e" 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 100, then have the
exclamation mark to the appropriate place, then delete the two
original characters, then press ENTER. Use upper case

40
ENTER the first command now

ERROR - you should type an EDIT 10 first!

ENTER the first command now EDIT 10

10 FOR I = 1 TO 10 PRINT I; NEXT I

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

1. FOR I = 1 TO 10 PRINT N; NEXT I

THEN ... if you want to do it again, press 'r' else press 'n'.
ENTER your choice now 'r' or 'n' 'n'

1. DELETE

For the example, we protected you from mistakes by ignoring 'r' 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 decide that you have done, the editor is very forgiving. It usually leaves you something, even when you make a fool of yourself.

As you can see itself, the 'r' switch in this title indicates that you are making 'r' characters at a time. If you press a letter and press 'r', then that is how many characters will be

pressed ENTER

10 FOR I = 1 TO 10 PRINT I; NEXT I

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

Note the quotation marks. In this case, are around both
characters. When you get through here, practice with some
***** Listing of Program " listing.TRT" *****

Lines you have arbitrarily saved, you will see that this
feature 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:

1. Text line

This is one of the easiest and most useful of the commands.
It allows you to start up at the end of a line, just as if you
never pressed ENTER.

First, you go to EDIT mode, then you press 1, when you do, you
will see the whole line displayed, and you can add editing
commands to it. Try it now. First, give the command to
set into EDIT mode (like this: then press 1, then type in
arrows of text until you want to modify allow a maximum of 249
characters). Then press ENTER. Do it now.

EDIT the first command? EDIT 1
1. FOR I = 1 TO K: PRINT I: NEXT I

Our new line, which was the sum of the old line 105 i.e. the
data you typed in. Let it test this:

THE 1 TO K: PRINT I: NEXT I

Now, you will find yourself using this command the most.

It usually starts you into saving characters, i.e. in:
allowing you to execute the \texttt{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 wipe out the end of the line you are editing, and replace the old data with new stuff. Be sure to practice this and you will grow to love it -- well, maybe just like it a lot.

press \texttt{ENTER}

which do you wish to do?

A Continue on
B Review this section again

dress the letter opposite the correct answer and press \texttt{ENTER} *A

\texttt{\textbf{A Change 1 I 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 numbers.

\texttt{FOR \texttt{K = \texttt{LET} PRINT \texttt{ I \texttt{NEXT} 1}}}

Give us our line (with the "*" characters missing -- we deleted them in the \texttt{LET} section). Let's say we want to change "\texttt{NEXT I}" to "\texttt{NEXT \texttt{ I} and \texttt{PRINT} \texttt{ I}}" to "\texttt{PRINT} \texttt{ I}"

In addition, we want the value of \texttt{I} to start at \texttt{1} in the \texttt{FOR \texttt{NEXT} 1} loop, i.e., we want to insert a \texttt{1 JUST before the col}\n
press \texttt{ENTER} to start our \texttt{EDITING} *
To use the NO(Change) command, get into the EDIT mode and position the cursor to JUST BEFORE the character to be changed, then press "O" and press the new character, then press ENTER. Try it now. Get to EDITor, SPACE over to just before the one you want to change, then type in "O" and type in an X we want to swap Y with it, and finally, press ENTER. Use capitals.

ENTER the first command at the bottom of the next line
10 FOR I = 1 TO 10 PRINT Y : NEXT I
EDIT 1:
10 FOR I = 1 TO 10 PRINT Y : NEXT 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?

ENTER a "Y" if you want to do this again, else ENTER an "N".

NOchange & Insert:

As with the other commands, the "N" in NO(Change) designates how many characters are affected by the command. If you want to change 10 characters, then you would type "10C" in the EDIT mode, and you would then HAVE to change the next 10 characters.

What would you type if you were in the EDIT mode and your cursor was just before a block of 4 characters that you wanted to change to "ABCD"?

ENTER your answer now: ADCC
EDIT Now you have the idea press ENTER

NOchange & Insert:

10 FOR I = 1 TO 10 PRINT Y : NEXT I
we would change the 'I' to an 'X' in the same way, but to
save time, we'll endure some hassle and change it now so we can
get to the INSERT command. READY? 

... R00F .. There, it's changed now. Look below.

10 FOR x = to77 : PRINT x : NEXT x

Now would you like to have that editor at your command?
We'll now get to the INSERT command. Remember, we want to
insert a 'Z' just before the 'to77'.
press ENTER for the insert example:

'Insert' and INSERT:

10 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
to insert the data in front of.

In this case, we get into the EDIT mode, then
we SPACE over to just before the 'to77' and then we type an 'I'
for INSERT. After the 'I' command we want to put in a 'Z',
but we C0ULD 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:

Repeat. First type EDIT 10, then space over to just before
the 'to77', then type 'Z', then type 11, then type ENTER.

10 FOR x = to77 : PRINT x : NEXT x

ENTER the first command EDIT 10

10 FOR x = to77 : PRINT x : NEXT x
***** Listing of Program 'LESSON16.TXT' *****

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

nChange: & (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 your nC or I command to fix up the line but I feel pretty magical today, so I want to do it.

ccccccccccccccc: FOOD ... Investigation...

Let's leave it up to you to do in your practice sessions.

press ENTER

Which do you wish to do?

A Continue on
B Review this section again

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

You have finished the lesson and you can now take the test.
If you wish to review parts of the lesson, ENTER an 'R'
else, if you want to continue to the test ENTER a 'C'

ENTER your choice now 'R' or 'C' or 'C'

Going to test. Please wait one moment.

FINAL TEST: Lesson 6
This test consists of 19 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$ : 8$ = LEFT$(B$,11) : PRINT B$

4 Multi Statement lines are not allowed
3 The assignment of values between A$ and B$ are not valid.
2 The strings were not initialized
1 Nothing

ENTER the letter opposite the correct answer> D

CORRECT

Press ENTER

10 A$ = "Hi"; B$ = A$ : 8$ = LEFT$(B$,11) : 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

If A$ = Hi and B$ = 4$, then the left character of B$ is an H. See part 1.

Press ENTER
***** Listing of Program 'LESSNO.TXT' *****

What is wrong with the following program.

10 FOR I = 1 to 20
20 A$(I) = "O"
30 NEXT I

* The array is not dimensioned properly
5 You cannot address a single dimensioned array with a loop
0 The "O" should not be enclosed in quotes
0 Nothing

ENTER the letter opposite the correct answer: a

WRONG - the answer is A (it should be DIMensioned to 20)
See part 1. String Arrays.

press ENTER?

What is the output of the following program?

10 A$ = "Hi!": B$ = A$: C$ = LEFT$(B$,1): PRINT B$,
20 D$ = A$ + B$

ENTER your answer EXACTLY as it would appear
CORRECT

press ENTER?

What is the output of the following program?

10 A$ = "SOMEBODY": B$ = MIDS(A$,1,4):PRINT A$

ENTER your answer EXACTLY as it would appear
SOME
CORRECT

press ENTER?
what is the command you would enter to edit line number 50 of a program?

ENTER your answer EXACTLY as it would appear? EDIT 50

CORRECT

press ENTER?

Assume you are in the EDIT mode. You wish to place the cursor over the second occurrence of the letter R in your line.

What is the command you would use?

A. ISR
B. ZII
C. ZIR
D. ZIR
E. I spacebar R

ENTER the letter opposite the correct answer? A

CORRECT

press ENTER?

Assume you are in the EDIT mode.

What is the command you would use to insert text starting where your cursor is now.

ENTER the command now? I

CORRECT

press ENTER?
Assume you are in the EDIT mode.

What is the command you would use to drive the cursor to the end of the line you are currently editing? (The command is one letter long.

ENTER the command now. X

CORRECT.

PRESS ENTER.

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

A. Just think, now that you know how
B. Practice, practice, practice . . . and 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 BASIC. What should you do?

A. Just think, now that you know how
B. Practice, practice, practice . . . and 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? YES

You are FINISHED WITH THIS COURSE!!!
It was GREAT having you as a student - THANKS!
When you press ENTER you will be sent to the Menu from
where you can review other lessons or quit.

For homework, you may wish to change your inventory program so that it will handle string data. That way you can include the names of your furniture in your file. However, we leave that up to you! Hasta Luego!

Press ENTER to return to MENU?

Break in 4250
Ready
s,ctea"reset esp"
TABLE OF CONTENTS

| HOMEWORK FOR LESSON 2 | 483 |
| HOMEWORK FOR LESSON 3 | 485 |
| HOMEWORK FOR LESSON 4 | 487 |
| HOMEWORK FOR LESSON 4 | 489 |
1060 LISTING OF PROGRAM "HWZ" 1070

1070 0SUB 1520
1080 PRINT"Your lesson 2 assignment is to write a program that requests the"
1090 PRINT"user to INPUT a number. The program should check the number"
1100 PRINT"against numbers in a data statement to see if there is a"
1110 PRINT"match. If there is a match, then the program should tell"
1120 PRINT"the user. Use the RESTORE and GOTO statements to keep the"
1130 PRINT"program running. Save the program to disk."
1140 PRINT
1150 PRINT"You will use much of what we have learned when you have this"
1160 PRINT"program. If you feel a little confused by the directions,"
1170 PRINT"you can get an example of a program that satisfies the"
1180 PRINT"requirements by turning on your printer and selecting the"
1190 PRINT"LIST PROGRAM TO PRINTER option below."
1200 PRINT
1210 PRINT"Which do you want to do?"
1220 PRINT"RETURN TO MENU"
1230 PRINT"LIST PROGRAM TO PRINTER"
1240 PRINT"RUN THE PROGRAM"
1250 PRINT
1260 INPUT"Enter your choice:"
1270 IF N = "A" THEN RUN "MENU"
1280 IF N = "B" THEN PRINT"PRINT"Be sure to type in RUN when the printer is done."
1290 IF N = "C" THEN GOSUB 1520 PRINT"Be sure to type RUN when program ends." PRINT"INPUT"press
1300 ENTER to start:TIME RUN 1200
1310 GOTO 1140
1320 REM This program asks for a number, doubles the number to a
1330 REM data list, and then tells the user if the correct number
1340 REM was selected. It is not the only way to do the problem.
1350 REM
1360 REM Next line resets the data statement (see lesson 2, part 2)
1370 RESTORE
1380 REM Now we ask for the number. I use 1 to 20 to keep it simple!
1390 INPUT"Enter a number between 1 and 20 (enter -1 to quit)":IN
1400 IF N = -1 THEN GOTO 1500
1410 REM Now we read a data point to see if we match (See lesson 2, part 2)
1420 READ D
1430 DATA 1,2,3,10,15,100
1440 REM Above DATA is arbitrary. I just made a few points up
1450 REM Now we compare the data point with the inputted number (see part 2)
1460 IF N = D THEN PRINT"MATCH - There is a data point that equals":IN
1470 REM If we have a match, we should RESTORE our data and start over
1480 IF N = D THEN GOTO 1500
1490 REM If our data point = J, then we are out of points to check, so
1500 REM we should RESTORE the data, tell the user there was no match
1510 IF J = 1 THEN PRINT"Your number does not match any of the data.
1450   IF D = 0 THEN GOTO 1300
1460   REM IF D doesn't equal 0 and it doesn't equal the present data
1470   REM point, then we read ANOTHER data point and check it all again.
1480   GOTO 1250
1490   REM
1500   END
1510   REM the following subroutine clears the screen
1520   FOR Y = 1 TO 24
1530       PRINT
1540   NEXT Y
1550   RETURN
**** Listing of Program "HW7" ****

10 S006 T0
20 PRINT"For your lesson 3 homework assignment, write a program"
30 PRINT"that requests three numbers from the user, and puts the"n
40 PRINT"numbers in an array that has been dimensioned to 3."
50 PRINT"Finally, print the array to the screen so that the numbers"n
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 (1-7):"
190 IF T = "A" THEN PRINT "RETURN TO MENU"n
200 IF T = "B" THEN PRINT:PRINT"Be sure to type in RUN when the printer is on."n
210 IF T = "C" THEN S006 T0:PRINT"You must type in RUN when the program"
220 PRINT"ends":PRINT:INPUT"press ENTER to start":"S006 T0:RUN T0
230 DATA 1,2,3
240 REM This program requests three numbers from the user and puts the"n
250 REM numbers in 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)"n
290 REM
300 REM DIM A(3)"
310 REM
320 REM Then we use a FOR NEXT loop to ask for three numbers. This is or:
330 REM easy. If 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 I = 1 TO 3
370 INPUT"Enter a number":A(I)
380 NEXT I
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":n
450 PRINT
**Listing of Program "HW7"**

07/12/82 - 03:14:22

450 REM We can sort the numbers in any ways, but the best straight
460 REM forward is to use IF statements
470 REM
480 REM For this problem, it helps to figure out how many ways three
490 REM numbers can occur. (The answer is 6 - 123, 132, 213, 231, 312, 321)
500 REM That means you can do it with six IF statements. (See lesson 2, part 2)
510 REM
520 IF A(1) AND A(2) AND A(3) THEN PRINT A(1), A(2), A(3)
530 IF A(2) AND A(1) AND A(3) THEN PRINT A(2), A(1), A(3)
540 IF A(3) AND A(2) AND A(1) THEN PRINT A(3), A(2), A(1)
550 IF A(1) AND A(3) AND A(2) THEN PRINT A(1), A(3), A(2)
560 IF A(2) AND A(3) AND A(1) THEN PRINT A(2), A(3), A(1)
570 IF A(3) AND A(1) AND A(2) THEN PRINT A(3), A(1), A(2)
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 to 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
**** Listing of Program 'HW4' *****

10 GOSUB 310
20 PRINT "For your lesson 4 homework: improve the program you made in"
30 PRINT "lesson 3. Write the array to disk. Write another sall program"
40 PRINT "to read the data back from the disk. THEN print the data in"
50 PRINT "ASCENDING order."  
60 PRINT
70 PRINT "I have made a sample program that satisfies the requirements.
80 PRINT "Run it or list it to the printer to see how it works."
90 PRINT "If you choose choice B, be sure your printer is ready."  
100 PRINT
110 PRINT "Which do you want to do?"
120 PRINT
130 PRINT "A RETURN TO MENU"
140 PRINT "B LIST PROGRAM TO PRINTER"
150 PRINT "C RUN THE PROGRAM"
160 PRINT
170 INPUT "Enter your choice": T$  
180 IF T$ = "A" THEN RUN "MENU"
190 IF T$ = "B" THEN PRINT:PRINT "Be sure to type in RUN when the printer is done."
200 IF T$ = "C" THEN GOSUB 310:PRINT:PRINT "You must type in RUN when the program"
210 GO TO 110
220 REM This program requests three numbers from the user and puts the
230 REM numbers in a three socket array. It then prints the array to disk.
240 REM The second part of the program reads the data back from disk and
250 REM prints it in ASCENDING order.
260 REM
270 REM First, we DIMension the array (see lesson 3 part 2)
280 REM
290 DIM A(3)
300 REM
310 REM Then we open a file to print the data to disk (see lesson 4 part 1)
320 REM
330 OPEN "G:\I.\TEST"
340 REM
350 REM Then we use a FOR NEXT loop to ask for three numbers (this is only)
360 REM an way of doing this problem, you may want to do it differently).
370 REM see lesson 3 part 1 for info on FOR NEXT)
380 REM
390 REM I chose to write the array to disk as it was entered by the user
400 REM
410 FOR I = 1 TO 3
420 INPUT "Enter a number": A(I)
430 PRINT A(I)
440 NEXT I
**** Listing of Program "HW4" ****

450 REM
451 REM Now we close the file (see lesson 4 part 2)
452 REM
453 CLOSE
454 REM
455 REM We'll stop the program here and let the user know what happened
456 REM
457 PRINT
458 PRINT "The data is on disk. When you press ENTER the data will be"
459 PRINT "read back from disk and printed in order. Press ENTER to"
460 REM
500 REM Now we print a general heading so that our output looks a little
501 REM better.
502 REM
503 PRINT
504 PRINT "These are your numbers in ASCENDING order:"
505 PRINT
506 REM
507 REM Open the file for input and read the three data points into array A
508 REM
509 OPEN "I",""TEST"
510 FOR I = 1 TO 3
511 INPUT #1,A(I)
512 NEXT I
513 CLOSE
514 REM We can sort the numbers in many ways, but the most straight
515 REM forward is to use IF statements (see lesson 1 part 2)
516 REM
517 REM For this problem, it helps to figure out how many ways three
518 REM numbers can occur. (the answer is 6 - 100, 100, 100; 100, 100, 99;...)
519 REM That means you can do it with six IF statements.
520 REM
521 IF A(I) = A(2) AND A(I) = A(3) THEN PRINT A(I),A(2),A(3)
522 IF A(I) = A(2) AND A(I) = A(3) THEN PRINT A(I),A(2),A(3)
523 IF A(I) = A(2) AND A(I) = A(3) THEN PRINT A(I),A(2),A(3)
524 IF A(I) = A(2) AND A(I) = A(3) THEN PRINT A(I),A(2),A(3)
525 IF A(I) = A(2) AND A(I) = A(3) THEN PRINT A(I),A(2),A(3)
526 IF A(I) = A(2) AND A(I) = A(3) THEN PRINT A(I),A(2),A(3)
527 END
528 REM
529 REM The following subroutine clears the screen
530 FOR I = 1 TO 24
531 PRINT
532 NEXT I
533 END
Listing of Program 'HN5'

10 GOSUB 2270
20 PRINT"Your homework assignment for lesson 5 is GREAT! You know enough."
30 PRINT"about computers to make a useful program that could ease your"
40 PRINT"workload. Your homework assignment is:

50 PRINT
60 PRINT"Write a program that will write a record to disk of all the"
70 PRINT"stock numbers of furniture that are in your office. (only use"
80 PRINT"a few numbers to start, until you get it debugged!)"
90 PRINT
100 PRINT"Next, write a program that will let you add a record onto"
110 PRINT"the end of the sequential file of stock numbers."
120 PRINT
130 PRINT"Finally, write a program that will search through the"
140 PRINT"disk file for a specific STOCK CLASS of numbers and"
150 PRINT"print them out to screen. (a stock class is the first 4 digits"
160 PRINT"of the stock number)"
170 PRINT
180 INPUT"Press ENTER":I0
190 GOSUB 2270
200 PRINT"HINT:"
210 PRINT
220 PRINT"This assignment requires you to use a number that exceeds 7 digits."
230 PRINT"And whenever you use a number that long in BASIC, the computer"
240 PRINT"always changes it to scientific notation (see your manual). To"
250 PRINT"prevent that, you must declare the number that you write and read"
260 PRINT"from disk as double precision. The command for that is: DEFDBL (var)"
270 PRINT
280 PRINT"10 DEFDBL I"
290 PRINT"20 I = 123456789"
300 PRINT"30 PRINT I"
310 PRINT"RUN"
320 PRINT
330 PRINT"1.23456789 E+08"
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":I0
390 GOSUB 2270
400 PRINT"HINT 02:"
410 PRINT
420 PRINT"In lesson 4, part 2, there was an example of how to update a"
430 PRINT"sequential file. Here is a summary of that example."
440 PRINT
450 PRINT"1. OPEN the file for sequential input"
460 PRINT"2. OPEN another new file for sequential output"
470 PRINT"3. INPUT a data point from the old file."
480 PRINT"4. OUTPUT the data point to the new file."
490 PRINT"5. When the old file is empty, add your new data to the end"
500 PRINT"of the new file."
510 PRINT
520 PRINT"THIS IS NOT THE MOST EFFICIENT METHOD, BUT MOST STUDENTS FIND IT"
530 PRINT"THE EASIEST TO START WITH. IN THE EXAMPLE PROBLEM, I WILL SHOW "
540 PRINT"YOU HOW TO DO IT MORE EFFICIENTLY!"
550 PRINT
560 INPUT"press ENTER";T$
570 GOSUB 2270
580 PRINT"HINT #3:"
590 PRINT
600 PRINT"To find the STOCK CLASS of an 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"A RETURN TO MENU"
840 PRINT"B LIST PROGRAMS TO PRINTER"
850 PRINT
860 INPUT"Enter your Choice";T$
870 IF T$ = "A" THEN RUN"MENU"
880 IF T$ = "B" THEN PRINT"Be sure to type RUN when the printer stops":ILLIST 209-2240
890 GOTO 810
900 REM
910 REM This is the first program. It requests stock #s and writes 'em to disk.
910 REM First we declare our variable as double precision and then open a file
920 REM
930 DEFDBL X
940 INPUT "What is the name of the disk file you want to OPEN"; F$ 
950 OPEN "O", 1, F$
960 REM
970 REM Then we ask for stock numbers
980 REM
990 INPUT "Please enter a stock number (-1 to quit)"; X
1000 IF X < 0 THEN GOTO 1110
1010 REM Then put them on disk and go back for more
1020 REM
1030 PRINT X
1040 PRINT
1050 GOTO 1010
1060 REM
1070 END
1080 REM
1090 REM This is the second program. It reads in the file made by the
1100 REM first program, prints it back to a new file, and then lets you
1110 REM add data to the end of the new file.
1120 REM
1130 REM First, declare our variable as double precision and open the file
1140 REM
1150 DEFDBL X
1160 INPUT "What is the name of the old data file"; F$
1170 OPEN "I", 1, F$
1180 REM
1190 REM Then open the new file (note that there are different buffer $s)
1200 REM
1210 INPUT "What is the name of the new data file you want to create"; N$
1220 OPEN "O", 2, N$
1230 REM
1240 REM Now read in the data from the old file and print it out to the new
1250 REM file until the old file is empty (see lesson 4 part 2, EOF)
1260 REM
1270 IF EOF(1) THEN GOTO 1390
1280 INPUT X
1290 PRINT X
1300 GOTO 1270
1310 REM
1320 REM When the old file is empty, the program jumps to the CLOSEI statement
1330 REM
1340 REM
1350 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
1570 REM This is the third program. It searches a disk file and tells you
1580 REM the number of occurrences of a stock class
1590 REM
1600 REM First, declare our variable as double precision and open a file
1610 REM
1620 DEFDBL X
1630 REM
1640 INPUT "What is the file of stock numbers that you want to search"; FS
1650 OPEN '"', 1, FS
1660 REM
1670 REM I've included an option to search for ALL the stock numbers
1680 REM
1690 INPUT "What is the stock class you are looking for (1 = ALL)"; N  
1700 PRINT
1710 PRINT "Here are the stock numbers in that class"
1720 PRINT
1730 IF EOF(1) THEN GOTO 1820
1740 INPUT #1, X
1750 IF N = 1 THEN PRINT X
1760 IF N = 1 THEN GOTO 1730
1770 IF INT (.0000001*X) = 0 THEN PRINT X
1780 GOTO 1730
1790 REM
1800 REM Once all the data has been checked, we close the file
1810 REM
1820 CLOSE
1830 PRINT
1840 INPUT "End of file. Do again (Y/N)?"; T$

492
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 INPUT#1,A(N) : "AS N INCREMENTS, ARRAY POCKETS ARE FILLED WITH DISK DATA
2030 SOTO
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#1,A(Y)
2180 NEXT Y
2190 CLOSE
2200 END
2210 REM
2220 REM "DIITA" WE DID ALL OUR EDITING WITH THE SAME FILE'
2230 REM
2240 REM
2250 REM this subroutine clears the screen
2260 REM
2270 FOR X = 1 TO 24
2280 PRINT
2290 NEXT X
2300 RETURN
APPENDIX E

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

10 REM This program started on 28 June 1983
20 REM Version number is 1 August 1983
30 REM
40 REM Author - Cast Danny J. Crespan. 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 T1 - T6 = temporary variables & counters
130 REM
140 CLEAR 10000
150 DIM N(50), S1(100), S2(100), S3(100), S4(100), S5(100), S6(100), S7(100), S8(100), S9(100)
160 ON ERROR GOTO 1880
170 * = o; 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
240 PRINT
250 PRINT
260 PRINT: PRINT: PRINT
***** Listing of Program 'REPORT' *****

470 LET N=1
480 INPUT #1, N$(N.524:1)
490 GOTO 480
500 CLOSE #1=OPEN #1", 'SCORES'
510 IF Y = 1 THEN GOTO 550
520 IF EOF(#1) THEN GOTO 550
530 X=X+1
540 INPUT #1, N$(X), S$(X)
550 GOTO 500
560 CLOSE #1=OPEN #1", 'SCORES'
570 IF X = 1 THEN GOTO 550
580 IF EOF(#1) THEN GOTO 550
590 J = J+1
600 INPUT #1, N$(J), S$(J)
610 GOTO 560
620 CLOSE #1=OPEN #1", 'SCORES'
630 IF Y = 1 THEN GOTO 550
640 IF EOF(#1) THEN GOTO 650
650 J = J+1
660 INPUT #1, N$(J), S$(J)
670 GOTO 610
680 CLOSE #1=1
690 REM
700 REM NOW WE SEARCH FOR DUPLICATE NAMES
710 REM
720 PRINT
730 PRINT 'Scores are read in, now I'm consolidating names "'
740 FOR N = 1 TO X
750 IF N+1 = CHRN(123) THEN GOTO 800: If leg already checked, then skip it
760 IF N+1 = N*Y THEN GOTO 700: If name is duplicate, call subroutine
690 NEXT N
770 PRINT '"
780 IF Y=1
790 IF X > 1 THEN GOTO 370
800 REM
810 REM WE CAN SORT THEM HERE, BUT IT TAKES A WHILE SO
820 REM USES HAS THE OPTION TO GO ON WITHOUT A SORT
830 REM
840 REM "The computer can sort out the names if you wish, however."
***** Listing of Program "REPORT" *****

760 PRINT"BASIC sorts take a few moments.";PRINT";PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;PRINT;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
*** Listing of Program 'REPORT' ***

1240 FOR X = 1 TO 14
1250 IF N$(X) = CHR$(124) THEN GOTO 1450
1260 IF LEN(N$(X)) = 0 THEN GOTO 1450
1270 LPRINT N$(X);:
1280 LPRINT TAB(20);: IF S1(X) = 0 THEN LPRINT "***";ELSE LPRINT S1(X);:
1290 LPRINT TAB(20);: IF S2(X) = 0 THEN LPRINT "***";ELSE LPRINT S2(X);:
1300 LPRINT TAB(20);: IF S3(X) = 0 THEN LPRINT "***";ELSE LPRINT S3(X);:
1310 LPRINT TAB(20);: IF S4(X) = 0 THEN LPRINT "***";ELSE LPRINT S4(X);:
1320 LPRINT TAB(20);: IF S5(X) = 0 THEN LPRINT "***";ELSE LPRINT S5(X);:
1330 LPRINT TAB(20);: IF S6(X) = 0 THEN LPRINT "***";ELSE LPRINT S6(X);:
1340 LPRINT
1450 NEXT X
1460 REM
1470 REM FOLLOWING LINE CAUSES FORMFEED ON MOST PRINTERS
1480 REM YOU MAY HAVE TO CHANGE IT FOR YOUR MACHINE
1490 REM
1500 LPRINT CHR$(140)
1510 PRINT "END OF PROGRAM - PRESS BREAK OR CONTROL C TO END ;"
1520 GOTO 1520
1530 REM
1540 REM THIS ROUTINE CONSOLIDATES DUPLICATE NAMES
1550 REM
1560 IF S1(X)=S1(Y) THEN S1(Y)=S1(X)
1570 IF S2(X)=S2(Y) THEN S2(Y)=S2(X)
1580 IF S3(X)=S3(Y) THEN S3(Y)=S3(X)
1590 IF S4(X)=S4(Y) THEN S4(Y)=S4(X)
1600 IF S5(X)=S5(Y) THEN S5(Y)=S5(X)
1610 IF S6(X)=S6(Y) THEN S6(Y)=S6(X)
1620 N$(X) = CHR$(124); "replace the duplicate name with a flag"
1630 RETURN
1640 REM
1650 REM THIS IS ERROR TRAP - IT PREVENTS PROGRAM STOPPING
1660 REM WHEN ONE OF THE SCORE FILES IS NOT FOUND
1670 REM
1680 PRINT "FILE ":Y: NOT FOUND"
1690 R = Y
1700 RESUME NEXT
1710 REM
1720 REM THIS CLEAR THE SCREEN ON MOST MACHINES
1730 REM
1740 FOR X = 1 TO 24
1750 PRINT
1760 NEXT X
1770 RETURN
1780 REM
1790 REM this is the bubble sort used to put the names in order
**** Listing of Program 'REPORT' ****

1800 REM
1810 I = X-1
1820 PRINT
1830 PRINT "Sorting through the names now!:
1840 FOR J=1 TO I
1850 IF N%(L)=N%(J) THEN GOTO 1980
1860 REM
1870 REM save first value
1880 REM
1890 T=Z%(L): T1=S1%(L): T2=S2%(L): T3=S3%(L): T4=S4%(L): T5=S5%(L): T6=S6%(L)
1900 REM
1910 REM swap array segments
1920 REM
1960 NEXT L
1970 NEXT J
1980 PRINT "":
1990 PRINT
2000 PRINT
2010 RETURN
REFERENCES CITED


18. Strategic Air Command Regulation 300-4

19. Tactical Air Command Regulation 300-12
