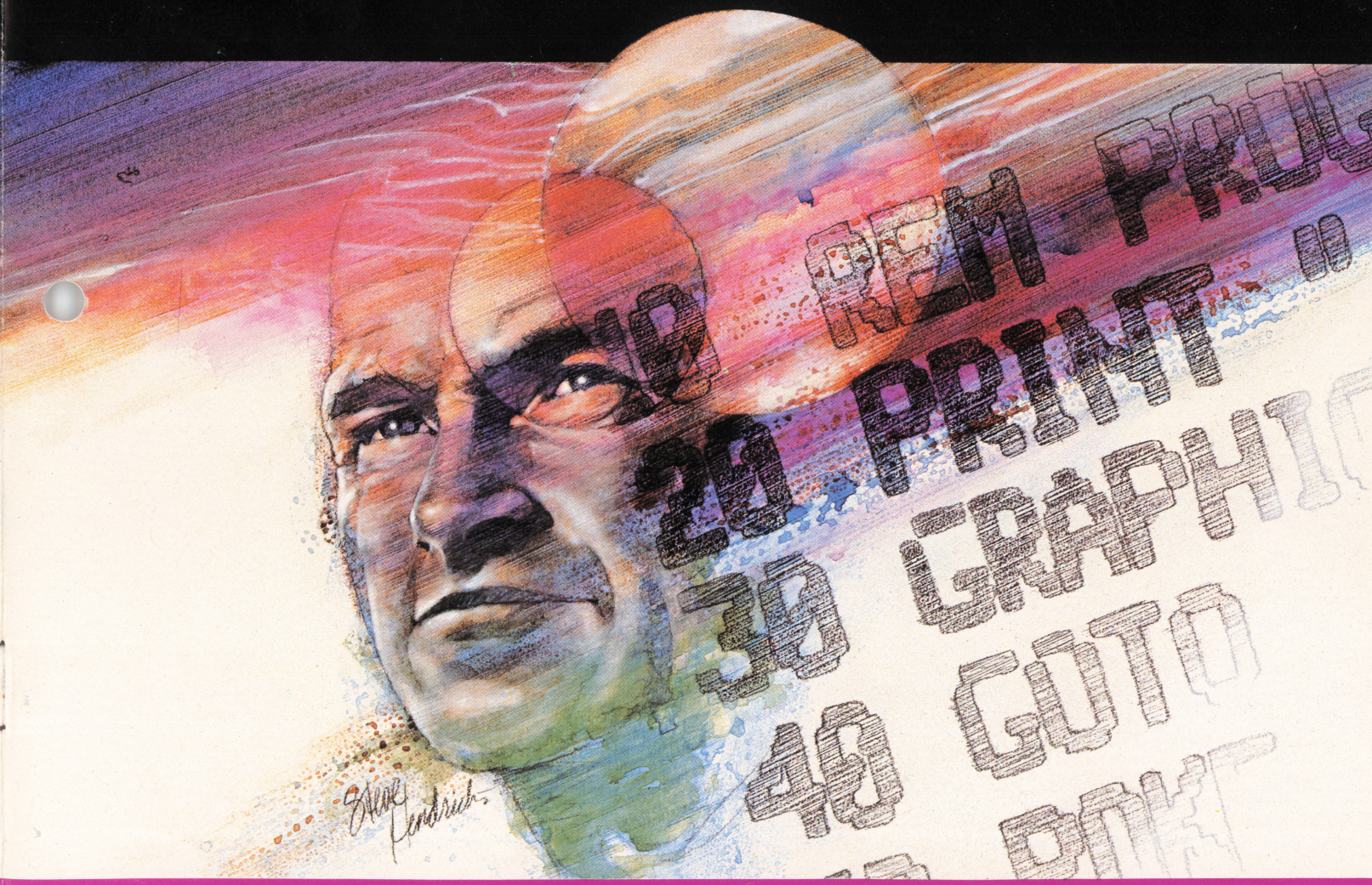
ATARI® 400/800™

AN INVITATION TO PROGRAMMING™ 2:

WRITING PROGRAMS ONE AND TWO



ATARI®

A Warner Communications Company

Model CX4106
Use with
ATARI® 400™ or ATARI 800™
PERSONAL COMPUTER SYSTEMS

AN INVITATION TO PROGRAMMING™ 2: Writing Programs One and Two

A Warner Communications Company

Every effort has been made to ensure that this manual accurately documents this product of the ATARI Computer Division. However, because of the ongoing improvement and update of the computer software and hardware, ATARI, INC. cannot guarantee the accuracy of printed material after the date of publication and cannot accept responsibility for errors or omissions.

Reproduction is forbidden without the specific written permission of ATARI, INC., Sunnyvale, CA 94086. No right to reproduce this document, nor the subject matter thereof, is granted unless by written agreement with, or written permission from the Corporation.

PRINTED IN U.S.A.

MANUAL AND PROGRAM CONTENTS © 1981 ATARI, INC.

CONTENTS

1 INTRODUCTION 1

2 HOW TO USE THIS COURSE 3

ATARI Components Required 3
General Instructions for Loading the Cassette Tape 3
Preparation for the Practice Sessions 5

3 SUMMARY OF LESSONS: WRITING PROGRAMS ONE 7

4 DESCRIPTION OF FRAMES FOR WRITING PROGRAMS ONE 9

Lesson One: Keyboard 9
Lesson Two: Printing on the Television Screen 9
Lesson Three: Characteristics of a Program 11
Lesson Four: Names and Numbers in Memory 15
Lesson Five: Computer Logic 19
Lesson Six: Library Function 23
Lesson Seven: FOR/NEXT Loops 25
Final Quiz: Writing Programs One 28

5 SUMMARY OF LESSONS: WRITING PROGRAMS TWO 29

6 DESCRIPTION OF FRAMES FOR WRITING PROGRAMS TWO 31

Lesson One: Screen Formatting Instructions 31
Lesson Two: READ/DATA 33
Lesson Three: Arrays 35
Lesson Four: PEEK and POKE 38
Lesson Five: ATASCII Code and Instructions 40
Lesson Six: String Handling 43
Lesson Seven: Multiple Statements 46
Lesson Eight: Subroutines 48
Final Quiz: Writing Programs Two 51

1

INTRODUCTION

AN INVITATION TO PROGRAMMING™ 2 covers beginning through advanced BASIC programming techniques. This course provides a two-part format with an interactive style of presentation, allowing you to experience immediate use of your computer. Completion of **WRITING PROGRAMS ONE** supplies you with rudimentary skills which can be applied in extending your knowledge of BASIC through actual "hands-on" application in **WRITING PROGRAMS TWO**.

The course has the following components:

- Cassette for Writing Programs One containing both the computer program and audio material
- Cassette for Writing Programs Two containing both the computer program and audio material
- This workbook containing both sections of the course to be used as a study guide and reference source

2

HOW TO USE THIS COURSE

ATARI COMPONENTS REQUIRED

- ATARI® 400™ or ATARI 800™ Personal Computer System with:

  8K (minimum) Random Access Memory (RAM)
  ATARI BASIC Computing Language Cartridge
- ATARI 410™ Program Recorder
- Writing Programs One Program Cassette
  Writing Programs Two Program Cassette

GENERAL INSTRUCTIONS FOR LOADING THE CASSETTE TAPE

Refer to Figure 1.

1. Connect the **ATARI 400** or **ATARI 800 Personal Computer System** to your television set and to a wall outlet as instructed in the Operator's Manual.

2. Make sure that the **ATARI 410 Program Recorder** is properly connected to the computer console and to a wall outlet (see your *ATARI 410 Program Recorder Operator's Manual* for further details, if necessary).

   Note: If you have "daisy chained" (connected in series) other ATARI peripherals to your computer console and do not wish to disconnect them, connect your ATARI 410 Program Recorder to the I/O CONNECTOR of the last unit in the chain.

3. Make sure that at least 8K of RAM is installed in your ATARI Personal Computer System. See the *Operator's Manual for Memory Module*™ loading instructions.

   Note: If an ATARI Disk Drive is connected to the computer console, the Disk Operating System (DOS) and system software use some of the available Random Access Memory (RAM). The amount of RAM used varies with the version of DOS you are using. Take this overhead into account when calculating the amount of RAM required to run a program.

4. Open the cartridge door and insert the ATARI BASIC cartridge into the cartridge slot. Use the **LEFT CARTRIDGE** slot on the ATARI 800 Personal Computer System. Close the cartridge door.

5. Turn on your television set. (You may have to adjust the volume on your television set, since the voice from the audio track comes from the television speaker.)

6. Turn on your ATARI Personal Computer System. The POWER switch is on the right side of the computer console.

7. If all equipment is properly connected and turned on, your television screen should display the READY prompt, with the white cursor just below.

8. Press **STOP/EJECT** on your Program Recorder to open the cassette door.

9. Load Lesson One by holding the cassette tape so that the label (Side 1) is up and the tape leader is facing you.

10. Slide the cassette into the cassette holder and close the door.

11. If necessary, press **REWIND** to rewind the tape to the beginning. (Set the counter on the Program Recorder to 000.)

12. Type **CLOAD** on the computer keyboard and press RETURN. The computer will "beep" once to remind you to press **PLAY** on the Program Recorder.

13. Press **PLAY** and the RETURN key to start the tape. Through the window in the Program Recorder, note that the tape is turning. The beeps and other sounds you hear coming from the television speaker tell you that Lesson One is being loaded into computer memory.

    Note: If you have problems loading either of the course programs and you have peripherals in addition to the Program Recorder attached to the computer console, try disconnecting the other peripherals and connecting the Program Recorder directly to the computer console to isolate any problem. If problems persist, consult the *ATARI 410 Program Recorder Operator's Manual.*

14. When the television screen displays the READY prompt, the first program encountered on the cassette, Lesson One, has been loaded into the computer. Make sure that the **PLAY** button on the Program Recorder is still down (and remains down). On the chart provided, make a note of the number on the Program Recorder's counter. Enter this number in the START AUDIO column for Lesson One. If you want to run Lesson One again, simply rewind the tape to this number. As you proceed with the course, similarly keep track of the START AUDIO numbers of the other lessons.

15. Type **RUN** and press RETURN to start Lesson One. From now on, follow the instructions given in the lesson. At the end of Lesson One, make a note of the Program Recorder counter number. Enter this number in the END AUDIO column for Lesson One.

16. **DO NOT** rewind the cassette tape. When you are ready to begin Lesson Two, follow the same procedure as you did to load Lesson One (begin with Step 12). Follow this procedure for all the lessons in the course and for the Final Quiz.

To go to a specific lesson in the course: Completely **REWIND** the tape. Set the counter at 000. Push **FAST FORWARD** on the Program Recorder. Advance the tape until you reach the START LOADING number that you entered on the chart.

To repeat a lesson just completed: **REWIND** the tape to the START AUDIO number on your chart.

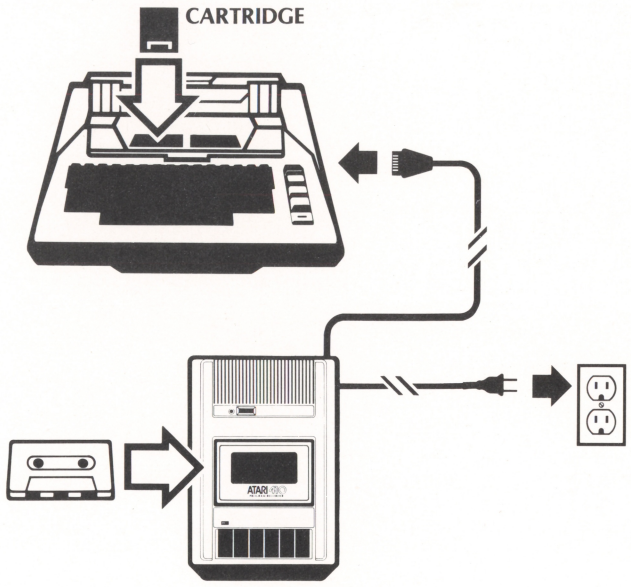


Figure 1 Loading a Cassette Tape

PREPARATION FOR THE PRACTICE SESSIONS

Memory in a computer system is divided into two components, ROM and RAM:

ROM. Read-Only Memory contains programs permanently stored in your computer by the manufacturer. This information is essential for computer operation. The ROM programs are called read-only because they can only be read, and they cannot be altered by use of the keyboard.

RAM. Random Access Memory temporarily stores programs and data in your computer. You can enter information directly into RAM from the keyboard, which allows you to create new programs, or bring stored data into your RAM workspace from diskette, cassette, or cartridge.

Turning off the computer console clears all the information stored in RAM. To write new data without turning your console off and on, use the BASIC instruction, **NEW** followed by RETURN. A **NEW** command clears any information in RAM.

A **CLOAD** command brings a stored program from the cassette into RAM. To clear out this program and prepare for the practice sessions, you MUST enter the **NEW** command.

The following statement is repeated at the beginning of each practice session because of its importance:

> **Note:** To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

3

SUMMARY OF LESSONS: WRITING PROGRAMS ONE

Lesson One: Keyboard
Reviews the ATARI computer console keyboard.

Lesson Two: Printing on the Television Screen
Covers the PRINT command for words or math in the direct operating mode. Includes the rules for procedural precedence in evaluating mathematical expressions.

Lesson Three: Characteristics of a Program
Explains BASIC program formatting and simple revisions. Gives initial exposure to the deferred operating mode by explaining a program RUN.

Lesson Four: Names and Numbers in Memory
Explains the manipulation of numeric and string variables.

Lesson Five: Computer Logic
Shows how to use IF/THEN and GOTO statements to allow for logical branching in the structure of a program.

Lesson Six: Library Function
Explains intrinsic functions available on the computer to get generation of random numbers, absolute or integer values, as well as square root computation.

Lesson Seven: FOR/NEXT Loops
Covers repetitive action using FOR/NEXT loops, the STEP instruction, and nesting of FOR/NEXT loops.

Final Quiz: Writing Programs One
A comprehensive review of the material covered in the course.

WRITING PROGRAMS ONE	LESSON	CASSETTE RECORDER COUNTER VALUE		
		START LOADING	START AUDIO	END AUDIO
SIDE 1	1	000		
	2			
	3			
	4			
SIDE 2	5	000		
	6			
	7			
	FINAL TEST			

Figure 2 Program Recorder Counter Numbers for Lessons in Writing Programs One

4

DESCRIPTION OF FRAMES FOR WRITING PROGRAMS ONE

LESSON ONE: KEYBOARD

The first lesson is a quick review of the ATARI keyboard. It includes an explanation of the following important keys:

- Backspace key (DELETE BACK S)
- Capital letter/lower case key CAPS LOWR
- Quotation mark key ''
- ATARI symbol key for reverse letters
- Break, control, escape keys (BREAK , CTRL , ESC)
- Math keys (+, -, /, *)

When you are ready, load Lesson Two on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON TWO: PRINTING ON THE TELEVISION SCREEN

Frame 1: Summary of Lesson Two

1. **PRINT** causes a display on the screen of whatever text or numeric characters you have placed within quotes.

2. **PRINT** is always in capital letters. Material you put inside the quotes can be upper or lower case, alpha or numeric characters.

3. **PRINT** used without quotes performs mathematical functions according to the following rules:

   The computer performs the math computation by doing:

   - Multiplication and division inside parentheses first.
   - Addition and subtraction inside parentheses next
   - Multiplication and division outside parentheses next
   - Addition and subtraction last.

On each step, the computer moves from left to right across the expression.

Example: (4/2+1) * 3 + (2+2) - 12

Step 1: (2+1) * 3 + (2+2) - 12
Step 2: 3 * 3 + 4 - 12
Step 3: 9 + 4 - 12
Step 4: 1

Computer prints: 1

Frame 2: Practice Session for Lesson Two

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

Enter each line on the computer by typing the command. Press RETURN to see the result. Try to determine beforehand what will be displayed on the screen and see if you're right.

```
PRINT "HELLO"
PRINT "MY NAME IS GEORGE"
PRINT "OH"
PRINT "Oh boy!"
print "OOPS"
PRINT "OOPS"
PRINT "@#$%&"
PRINT "THIS IS RIGHT"
```

Try this math. Enter each command and press RETURN.

```
PRINT "3-3"
PRINT "7*7"
PRINT 4*4
PRINT 10/2+4-5
PRINT (3*6)/(3*3)
PRINT 3*6/3*3
PRINT 25+(6/2-2)-8
PRINT 200/(3+2)*2-60
PRINT (6-2)*2*5
```

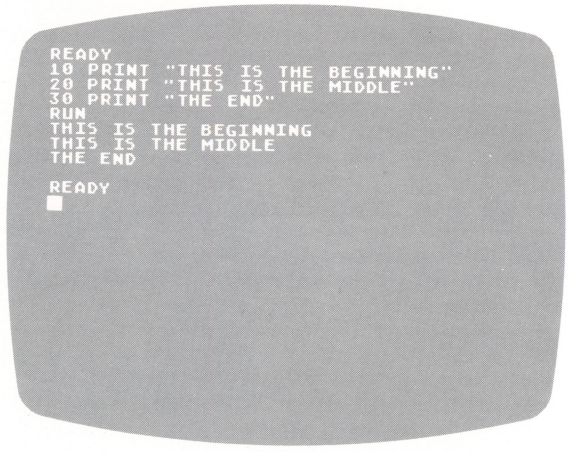
Now use the computer to solve these problems:

- Add three numbers and PRINT the result (pick the numbers).
- Compute the average of five numbers. (Add them and divide by 5. Remember the operational precedence of mathematical expressions when you enter this exercise into the computer. Placement of parentheses is emphasized in this case.)

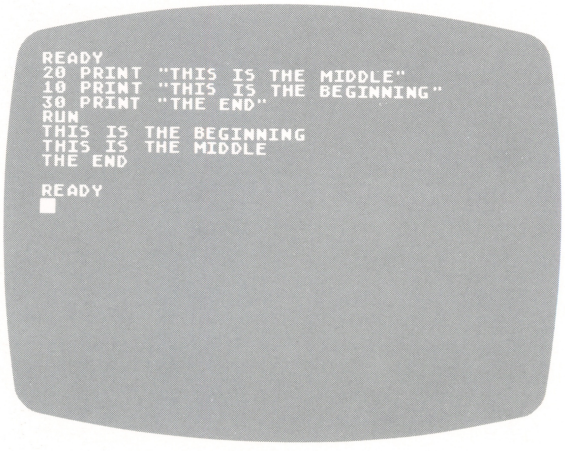
When you are ready, load Lesson Three on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON THREE: CHARACTERISTICS OF A PROGRAM

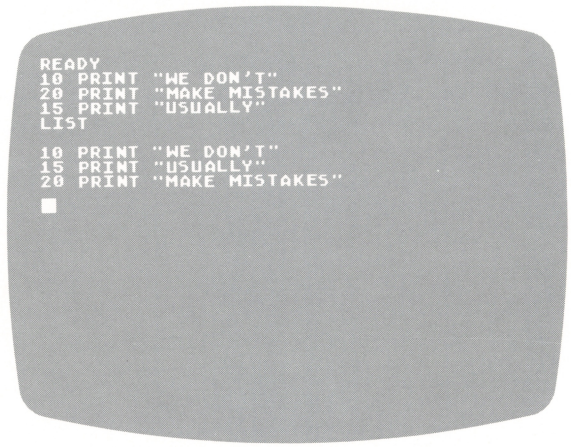
Frame 3: Entering a Program



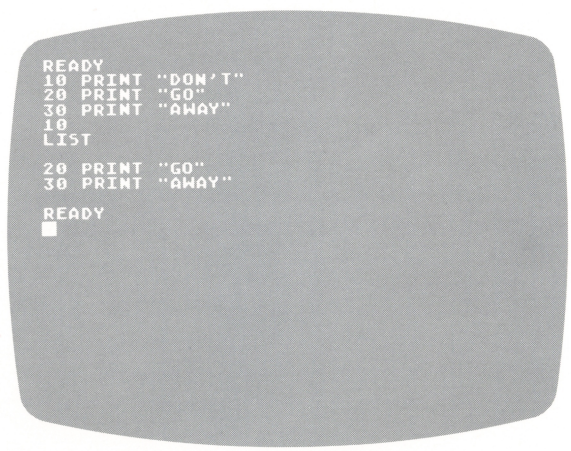
Frame 4: Example of Executing Consecutive Line Numbers



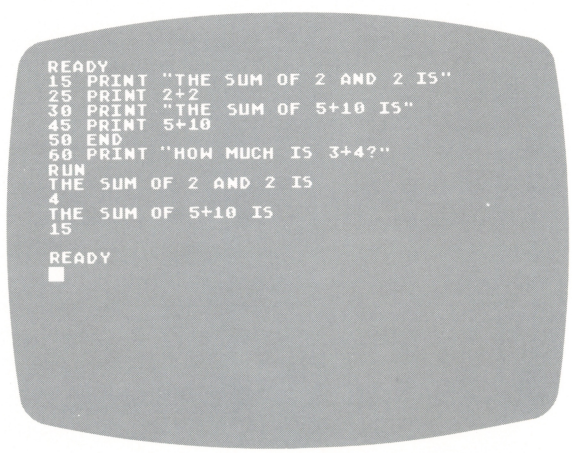
Frame 5: Insertion of Program Line



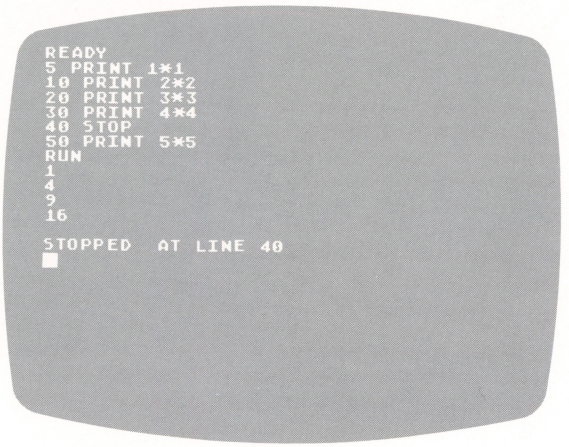
Frame 6: Deletion of Program Line



Frame 7: Use of END Command



Frame 8: Use of STOP Command



Frame 9: Summary of Lesson Three

1. A program is a series of logical steps for the computer. Each step gets a unique line number.

2. **RUN** tells the computer to execute the lines in consecutive numerical order.

3. **LIST** tells the computer to display all the program lines in consecutive numerical order.

4. **NEW** erases the program from computer memory.

5. **END** or **STOP** statements cause the computer to terminate program execution at the line number of the command.

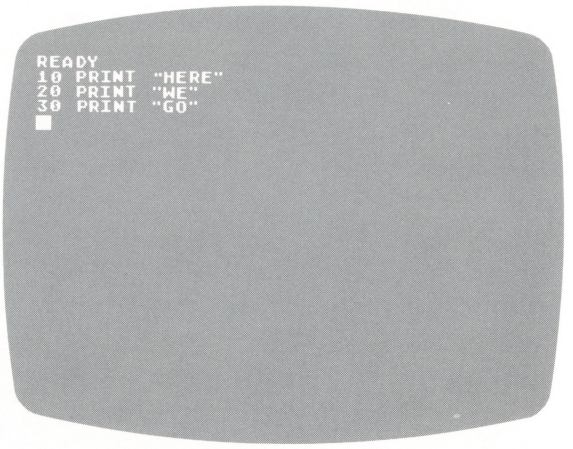
6. To edit your program

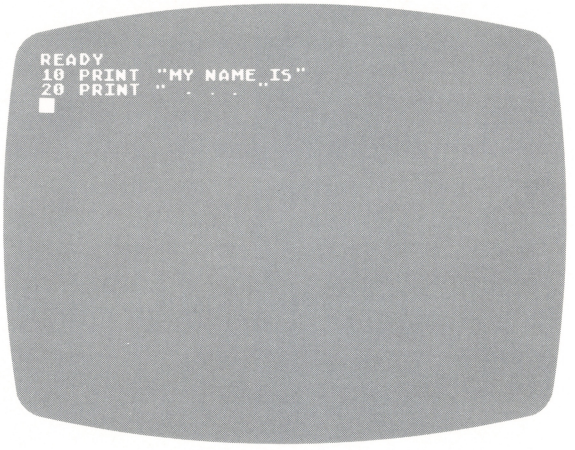
   - Retype the line.
   - Add a new line with a line number that is numerically between other lines (25 placed between line numbers 20 and 30), or
   - Remove a line by typing its number and pressing RETURN.

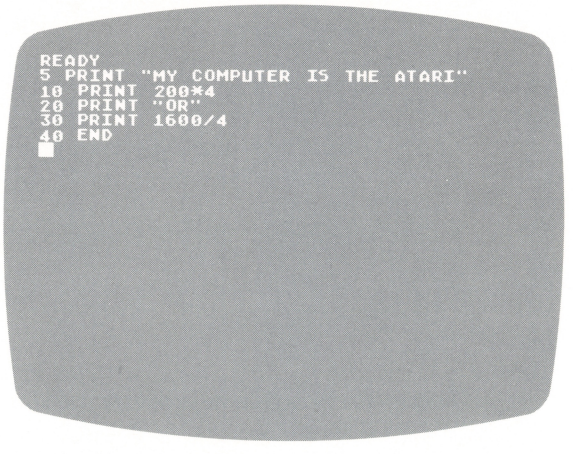
Frame 10: Practice Session for Lesson Three

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

Enter and run each of these programs.







Use the core program above. Add the following additional program lines one at a time to the basic program. After you add each line, run the program to see the effect of the change.

```
READY
25 PRINT "THE ATARI"
30 PRINT "FOUR HUNDRED"
50 PRINT "THIS WON'T PRINT"
15 PRINT
```

When you are ready, load Lesson Four on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON FOUR: NAMES AND NUMBERS IN MEMORY

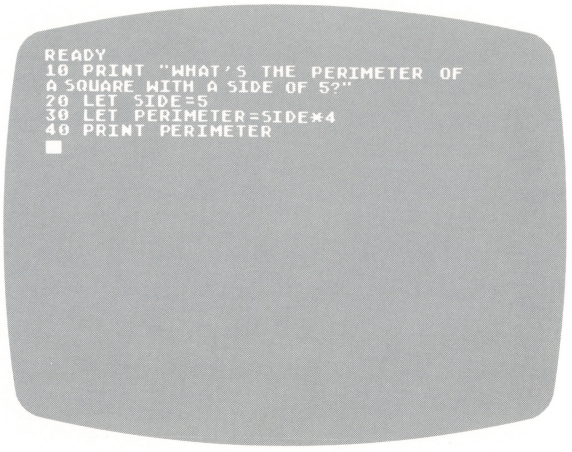
Frame 11

```
READY
10 PRINT "PROGRAM TO COUNT STARS"
20 LET BIGDIPPER=7
30 LET LITTLE=7
40 LET DOGSTAR=1
50 LET STARS=BIGDIPPER+LITTLE+DOGSTAR
60 PRINT STARS
70 PRINT "OOO, A FALLING STAR!"
80 LET STARS=STARS+1
90 PRINT STARS
```

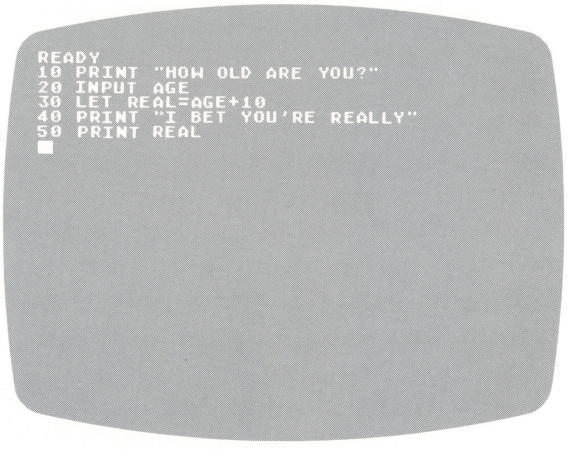
Frame 12

```
READY
5 LET D=1
10 LET D=D+1
20 LET D=D+1
30 LET D=D+1
40 PRINT D
```

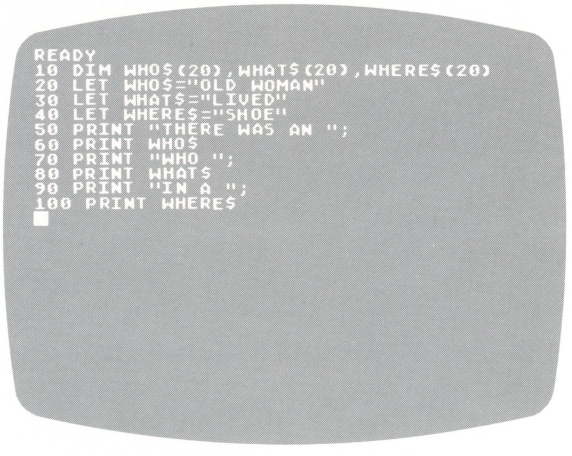
Frame 13



Frame 14



Frame 15



Frame 16: Summary of Lesson Four

1. Variables identify memory locations in which the computer stores specific alpha or numeric characters, for example, the result of a computation or an assigned value.

2. Variable types determine the kind of labels required.
   - Numeric variables reference numbers ONLY.
   - String variables store any assigned sequence of alpha or numeric characters or their combination.

   Each item of information is given a name, which will be used throughout the program to refer to that particular item. This name is called an identifier.

3. Specific rules for variable labels include:
   - Assign identifiers like A, B, C3, DE49 to numeric variables.
   - Assign identifiers like A$, B2$, ERROR$ to string variables.
   - DO NOT use reserved BASIC words as labels, such as: FOR, NEXT, DIM, LET, PRINT.

4. Commands for variable operation include:
   - **LET** stores numbers or letters into a variable.

     LET X=100
     LET X$= "SHIRLEY"
   - **DIM** signifies a maximum amount of characters allowed into a variable.

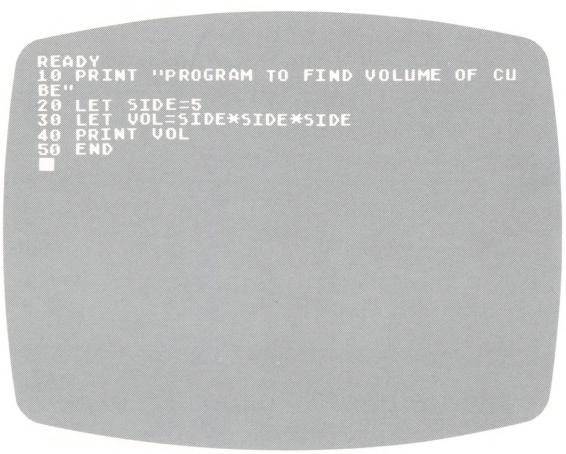
     DIM A$(10)
   - **INPUT** commands the computer to wait for entry of the number or word to be stored in the numeric or string variable.

     INPUT X
     INPUT X$ (remember to DIM X$ first)

Frame 17: Practice Session for Lesson Four

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

Run each of these sample programs.

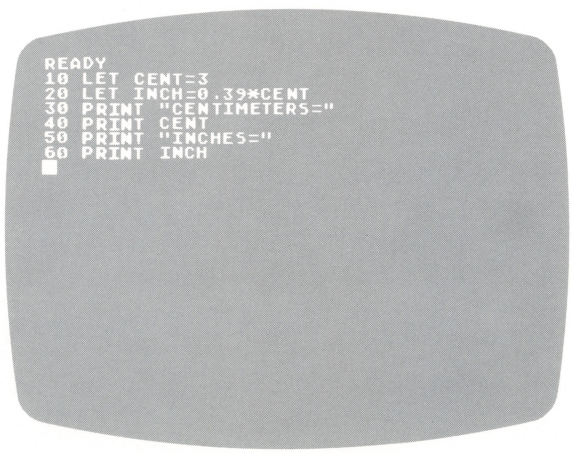


Add these lines to make an expanded program.

```
32  PRINT "SIDE=";
34  PRINT SIDE
36  PRINT "VOLUME=";
```

THEN change Line 20 to find the volume of other cubes.

The following program converts centimeters to inches.



Now change Line 10 and run the program several times, typing in a different number each time.

```
10 INPUT CENT
```

Add this line and RUN AGAIN.

```
5 PRINT "HOW MANY CENTIMETERS"
```

Type **NEW** and try this program.

```
READY
10 DIM NAME$(20),MONTH$(20)
20 PRINT "WHO ARE YOU?"
30 INPUT NAME$
40 PRINT "WHAT MONTH WERE YOU BORN?"
50 INPUT MONTH$
60 PRINT "HELLO ";
70 PRINT NAME$
80 PRINT "THE NICEST PEOPLE ARE BORN I
N"
90 PRINT MONTH$
```

When you are ready, load Lesson Five on the cassette. You must turn over the cassette, Writing Programs One, to Side 2. Refer to General Instructions for Loading the Cassette Tape. Remember to reset the Program Recorder counter to 000.

LESSON FIVE: COMPUTER LOGIC

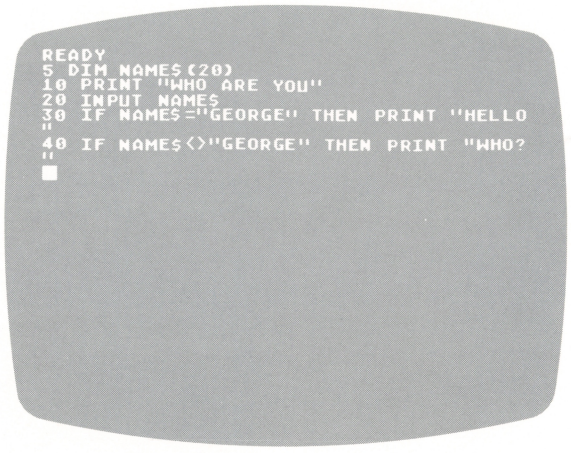
Frame 18

```
READY
10 PRINT "WHAT IS 3+3?"
20 ANSWER=6
30 INPUT RESPONSE
40 IF RESPONSE=ANSWER THEN PRINT "CORR
ECT"
50 END
RUN
WHAT IS 3+3?
?
```

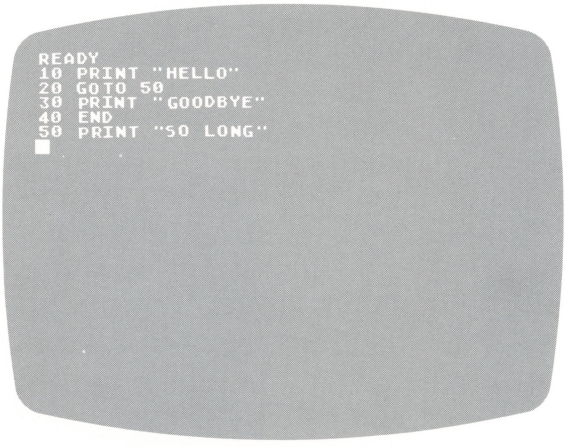
Frame 19

```
READY
10 PRINT "HOW OLD ARE YOU?"
20 INPUT AGE
30 IF AGE>17 THEN PRINT "YOU CAN VOTE"
40 IF AGE<18 THEN PRINT "YOU CAN'T
VOTE"
```

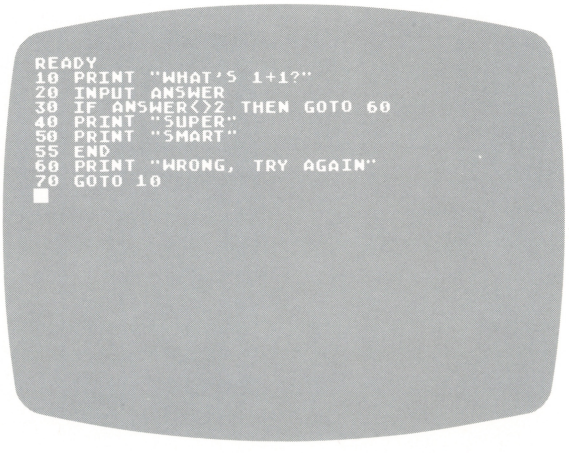
Frame 20



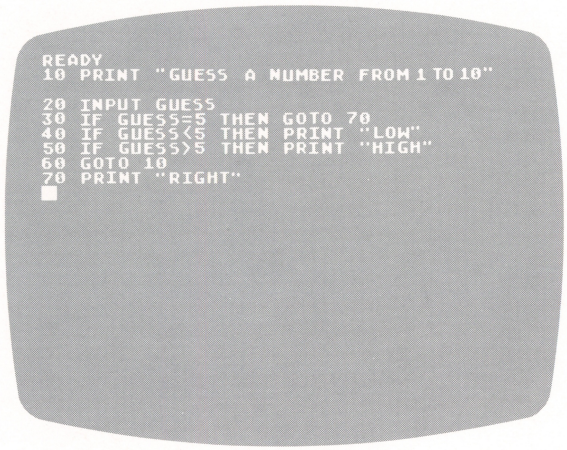
Frame 21



Frame 22



Frame 23



Frame 24: Summary of Lesson Five

1. **IF/THEN** program statements provide for conditional logic sequencing. If a stipulated condition holds true, the computer performs the instruction present in the **THEN** command. If the condition proves false, the computer ignores the **THEN** portion of the statement and goes to the next program line.

2. Logical Operators are:

   = means "equals"

   An **IF/THEN** statement comparison between two string variables will only hold true if the strings prove equal, character for character.

   <> means "not equal"

   > means "greater than"

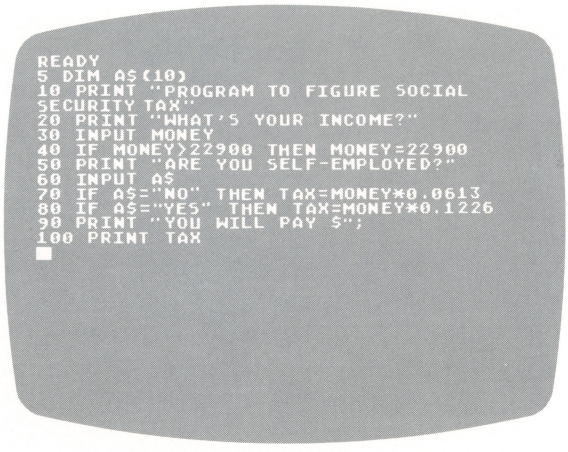
   < means "less than"

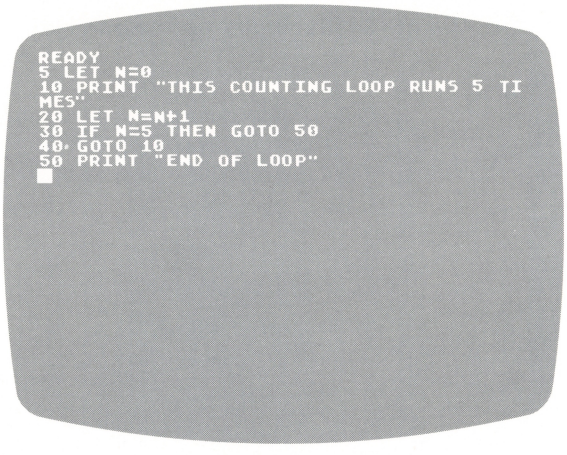
3. **GOTO** provides for an unconditional logic sequence. The computer moves to the specified line and continues processing.

4. Use the BREAK key to end computer execution of an infinite loop.

Frame 25: Practice Session for Lesson Five

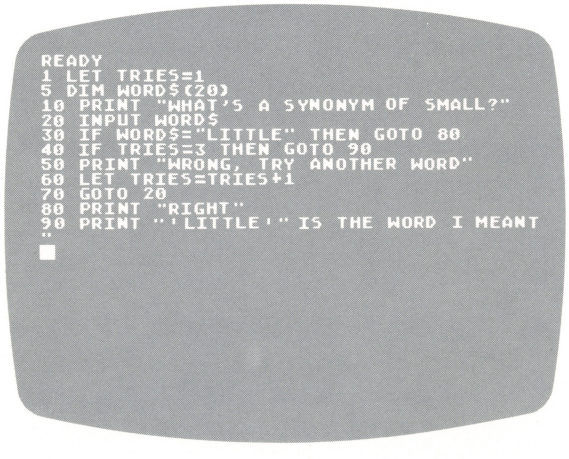
Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.





Change line 40 in the above program to GOTO 5. Why doesn't this work?

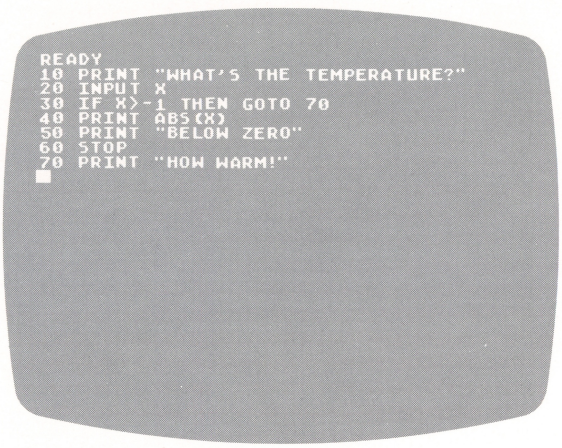
Run the following program and make three intentional mistakes to see how the loop ends.



When you are ready, load Lesson Six on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON SIX: LIBRARY FUNCTION

Frame 26



Frame 27

LOG(X) = Logarithm
SIN(X) = Sine
COS(X) = Cosine
SGN(X) = Sign—plus or minus

Frame 28: Summary of Lesson Six

1. Library functions perform their intended calculation based upon previously established internal programming. Therefore, the use of these functions allows you to "return" a number in the form you need for further computation in a calculation or storage in a variable.

2. Common library functions include:

   - **INT** rounds a number to its next lower whole number.
   - **ABS** gives you the value of a number without the + or - sign.
   - **RND** used with (0) gives a random number between 0 and 1.
   - **SQR** gives you the square root of a number.
   - **FRE** used with (0) tells how much free memory is available.

Frame 29: Practice Session for Lesson Six

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

Here's a program that tells whether a number is odd or even.

```
READY
10 PRINT "TYPE A WHOLE NUMBER."
20 INPUT N
30 IF N/2=INT(N/2) THEN PRINT "EVEN"
40 IF N/2<>INT(N/2) THEN PRINT "ODD"
50 END
```

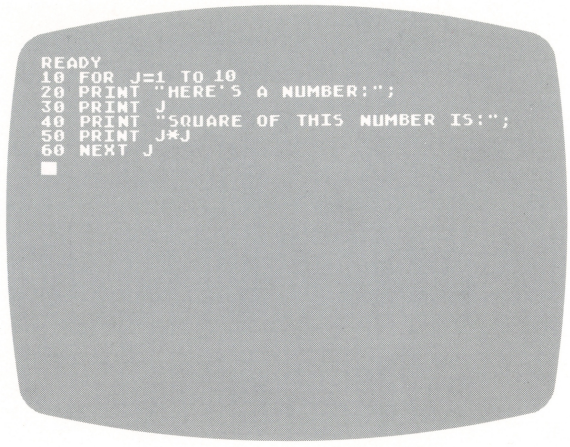
Try this program with different inputs. Then BREAK it.

```
READY
5 DIM NAME$(20)
10 PRINT "WHO ARE YOU?"
20 INPUT NAME$
30 X=RND(0)
40 Y=INT(3*X)
50 IF Y=0 THEN PRINT "WHAT KIND OF NAM
E IS THAT?"
60 IF Y=1 THEN PRINT "WHO!!?"
70 IF Y=2 THEN PRINT "NICE NAME"
75 PRINT
80 PRINT
90 GOTO 10
```

When you are ready, load Lesson Seven on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

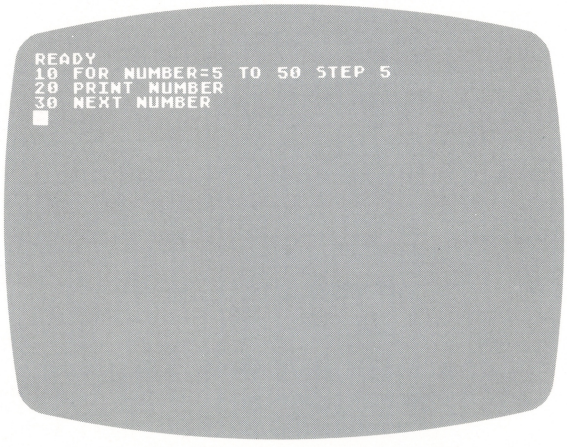
LESSON SEVEN: FOR/NEXT LOOPS

Frame 30

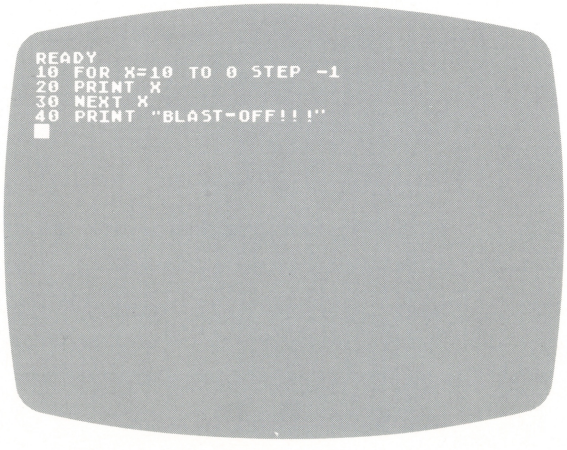


Program continues until loops are completed.

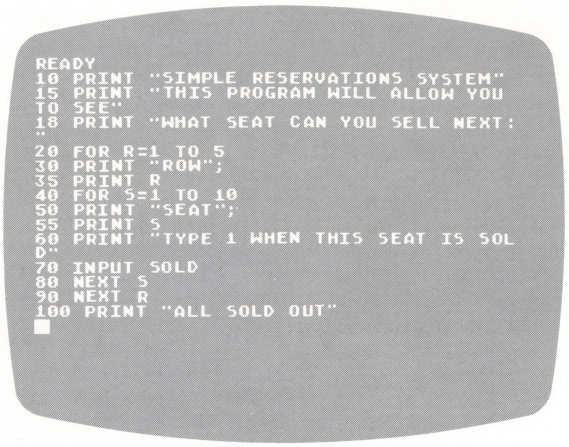
Frame 31



Frame 32



Frame 33



Frame 34: Summary of Lesson Seven

A **FOR/NEXT** counting loop repeats a program segment by executing those statements contained within the loop a given number of times.

Set a dummy variable X at 1 and end the loop after a specified number of increments.

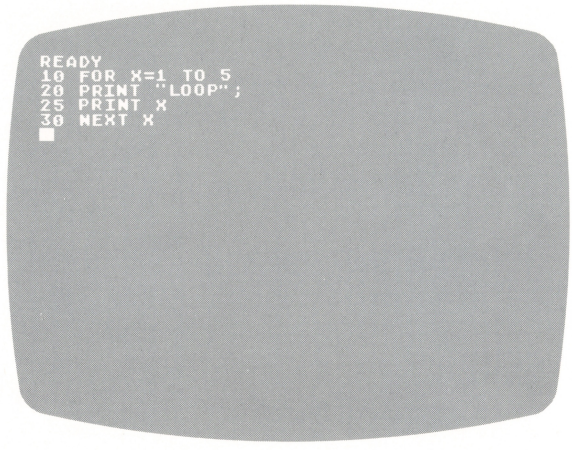
```
10 FOR X=1 TO 10
.
.
.
50 NEXT X
```

In the program above, line statement 50 tells the computer to return to the **FOR** statement line and process back down to line 50. Each time the computer encounters this program line, X increments by 1 and returns to the beginning of the **FOR/NEXT** loop until all program lines contained between the **FOR/NEXT** statements are executed. The "loop" process continues until the desired number of times is reached. At that point, execution of the program advances to the next consecutive numeric line statement.

Frame 35: Practice Session for Lesson Seven

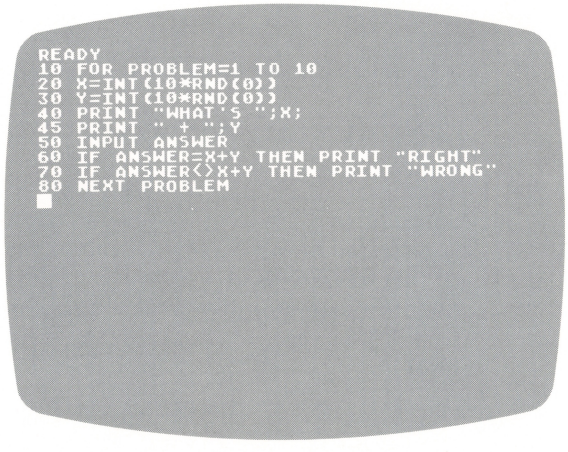
Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

Run the following program.



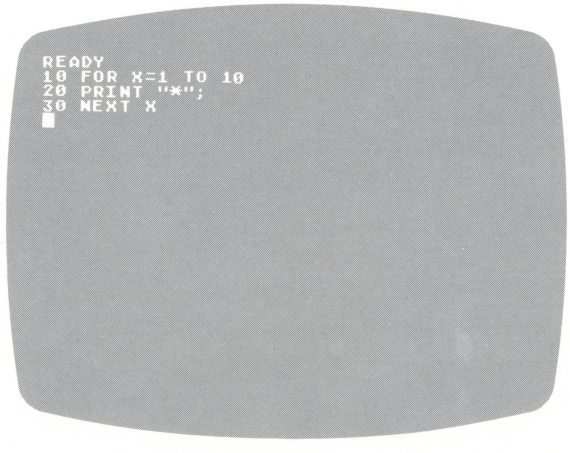
Change line 10 to get more loops.

The following program generates ten math problems.



Change the program so you get problems with larger numbers.

The next program prints a line of dots.



After running the program, add these lines to make a "nested" loop and run it again.

```
15  FOR Y=1 TO X
25  NEXT Y
26  PRINT
```

Then change this line.

```
10  FOR X=10 TO 1 STEP -1
```

Add these lines.

```
11  Z=37-X
12  FOR A=1 TO Z
13  PRINT " ";
14  NEXT A
```

Change the program to invert the triangle but leave it at the right.

CONGRATULATIONS! GO ON TO THE FINAL QUIZ ON THE CASSETTE.

Take the Final Quiz for Writing Programs One. The program will automatically calculate your scores after you complete the quiz. Load the Final Quiz in the same manner in which you loaded the other lessons. Proceed to the next section of this workbook by loading your cassette, Writing Programs Two, using the procedure outlined in General Instructions for Loading the Cassette Tape.

5

SUMMARY OF LESSONS: WRITING PROGRAMS TWO

Lesson One: Screen Formatting Instructions
Shows method for clearing screen and demonstrates how to position the cursor and print in columns.

Lesson Two: READ/DATA
Shows how to assign values to different variables through the use of READ and DATA statements.

Lesson Three: Arrays
Shows how to set up single-dimension arrays and how to use them in a program.

Lesson Four: PEEK and POKE
Explains computer memory and how to access it directly by means of PEEK and POKE statements.

Lesson Five: ATASCII Code and Instructions
Explains ASCII code as it relates to the ATARI Personal Computer System, and how to use the CHR$ and ASC functions.

Lesson Six: String Handling
Demonstrates how to examine the contents of string variables and how to use parts of the contents in logic statements.

Lesson Seven: Multiple Statements
Shows how to write lines with more than one statement. Explains logic flow when multiple statement lines are used with IF/THEN statements.

Lesson Eight: Subroutines
Explains the concept of subroutines and demonstrates the use of GOSUB and RETURN.

Final Quiz: Writing Programs Two
A review of the material covered in Writing Programs Two. No voice track is used with this part of the program.

WRITING PROGRAMS TWO	LESSON	CASSETTE RECORDER COUNTER VALUE		
		START LOADING	START AUDIO	END AUDIO
SIDE 1	1	000		
	2			
	3			
	4			
SIDE 2	5	000		
	6			
	7			
	8			
	FINAL TEST			

Figure 3 Program Recorder Counter Numbers for Lessons in Writing Programs Two

6

DESCRIPTION OF FRAMES FOR WRITING PROGRAMS TWO

LESSON ONE: SCREEN FORMATTING INSTRUCTIONS

Frame 1: Summary of Lesson One

1. **CURSOR**: Wherever the cursor appears on the screen, the computer will accept input and display the character received from the keyboard.

2. **GRAPHICS 0**: Clears the screen of text and sets the background color to blue. The cursor will automatically move to the top and extreme left of the screen, commonly called the "home" position.

3. **SEMICOLON**: Use of the semicolon in a PRINT statement keeps the output from moving down to the next line when text and numbers are printed on the screen.

   Example:
   ```
   10 DIM NAME$(20)
   20 NAME$="JOE"
   30 PRINT "MY NAME IS ";NAME$;"."

   MY NAME IS JOE.
   ```

4. **COMMA**: Working in a fashion similar to the semicolon, the comma places the printed text or numbers into zones on the screen.

   Example:
   ```
   PRINT "ONE","TWO","THREE"
   PRINT 1,2,3

   ONE          TWO          THREE
   1             2             3
   ```

Frame 2: Practice Session for Lesson One

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

The following program illustrates the use of the comma and the semicolon.

```
READY
5 GRAPHICS 0
6 PRINT
10 PRINT "LINE"
20 PRINT "SPACE"
30 PRINT
40 PRINT "LINE";
50 PRINT "SPACE"
60 PRINT
70 PRINT "LINE","SPACE"
80 END
```

Now run this program:

```
READY
5 GRAPHICS 0
6 PRINT
10 PRINT "TYPE IN TWO NUMBERS"
20 PRINT "FIRST NUMBER =";
25 INPUT A
30 PRINT "SECOND NUMBER =";
35 INPUT B
40 C=A+B
50 PRINT
60 PRINT A;"+";B;"=";C
70 PRINT
80 GOTO 10
```

This program illustrates the use of the comma to make columns.

```
READY
5 GRAPHICS 0
10 PRINT
20 PRINT "NUMBER","SQUARE","SQ.ROOT"
25 PRINT
30 FOR NUMBER=1 TO 10
40 LET SQUARE=NUMBER*NUMBER
50 LET ROOT=SQR(NUMBER)
60 PRINT NUMBER,SQUARE,ROOT
70 NEXT NUMBER
80 END
```

The next program illustrates how text can be strung together with semicolons. To view parts of the program **LIST** as follows:

LIST 10,50

This will list all lines between 10 and 50.

```
READY
5 DIM PLACE$(16)
6 DIM PART$(16)
7 DIM LIQUID$(16)
10 GRAPHICS 0
20 PRINT "TYPE IN THE NAME OF A PLACE
OR THING:"
30 INPUT PLACE$
40 PRINT "TYPE IN THE NAME OF A BODY P
ART:"
50 INPUT PART$
60 PRINT "NAME A LIQUID:"
70 INPUT LIQUID$
80 PRINT "HERE IS YOUR MADLIB:"
90 PRINT "JACK AND JILL"
100 PRINT "WENT UP THE ";PLACE$
110 PRINT "TO FETCH A PAIL OF";LIQUID$
120 PRINT "JACK FELL DOWN"
130 PRINT "AND BROKE HIS ";PART$
140 PRINT "AND JILL CAME TUMBLING AFTER!"
150 END
```

When you are ready, load Lesson Two on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

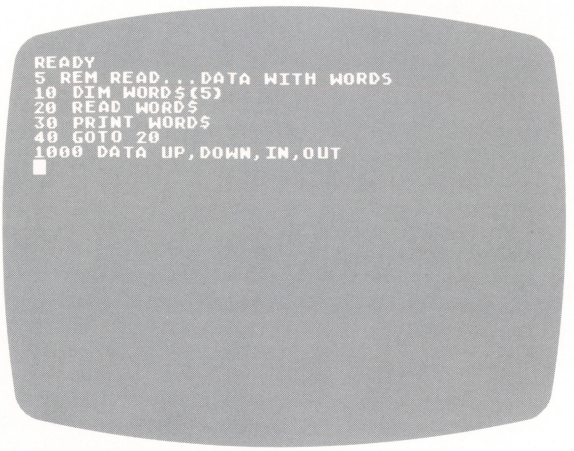
LESSON TWO: READ/DATA

Frame 3

The **READ** and **DATA** statements contained in the following program illustrate the method of execution used by the computer. When a **READ** statement is encountered, the first **DATA** statement is scanned by the computer. The assigned variable receives the initial value contained in the **DATA** statement. This value will remain constant until a new **READ** instruction is encountered.

```
READY
100 DATA 3,8,0,-4
200 PRINT
300 PRINT
400 READ N
405 DATA 53,4,9
410 PRINT N
420 GOTO 200
```

Frame 4



Frame 5: Practice Session for Lesson Two

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

```
10  READ A
20  PRINT A
30  GOTO 10
40  DATA 5, 6, 7, 8
```

Run the above program and then add the following line to it.

```
25  IF A=8 THEN RESTORE
```

Add the following lines:

```
5   LET B=0
26  LET B=B+1
27  IF B=8 THEN STOP
```

Add the following line.

```
6   DATA 1, 2, 3, 4
```

Add the following lines.

```
10  READ X, Y
20  PRINT X, Y
```

And delete 35, 36, and 37.

Run the following program:

```
READY
5 DIM INCORRECT$(10),CORRECT$(10),ANSW
ER$(10)
10 GRAPHICS 0
20 PRINT
25 PRINT
30 PRINT "CORRECT THE FOLLOWING MISSPE
LLED WORDS"
40 READ INCORRECT$,CORRECT$
50 IF CORRECT$="LAST" THEN END
60 PRINT
61 PRINT INCORRECT$
65 PRINT "THE CORRECT SPELLING IS: ";
70 INPUT ANSWER$
80 IF ANSWER$<>CORRECT$ THEN GOTO 50
90 IF ANSWER$=CORRECT$ THEN PRINT "COR
RECT!!"
100 GOTO 40
1000 DATA LIBARY,LIBRARY,SINTAX,SYNTAX
,NABOR,NEIGHBOR,LAST,LAST
```

When you are ready, load Lesson Three on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON THREE: ARRAYS

Frame 6: Summary of Lesson Three

1. Arrays are sets of variables with the same name. Each array variable has the name (in capitals) followed by a number in parentheses. If only one number is assigned, the array is one dimensional. Visualize the elements in the array being placed in one long column. For maximum speed and efficiency of operation, dimension your arrays into rows and columns. The first number within the parentheses indicates the assigned row, and the second number indicates the assigned columm in a two-dimensional array. Carry this concept into n-dimensions by simply adding rows and columns.

   Examples of one-dimensional arrays:

   A(10),VAR(5) or VAR(X) or ARRAY(NUM)

   Examples of two-dimensional arrays:

   A(10,50),VAR(5,5) or VAR(X,Y) or ARRAY(ROW,COLUMN)

2. Arrays must first be set up in ATARI BASIC with a **DIM** (dimension) statement.

   DIM VAR(10)

3. Values can be placed into and removed from array variables with simple **FOR/NEXT** loops.

   10 DIM ARRAY(10)

   20 FOR X=1 TO 10

   30 LET ARRAY(X)=25.2

   40 NEXT X

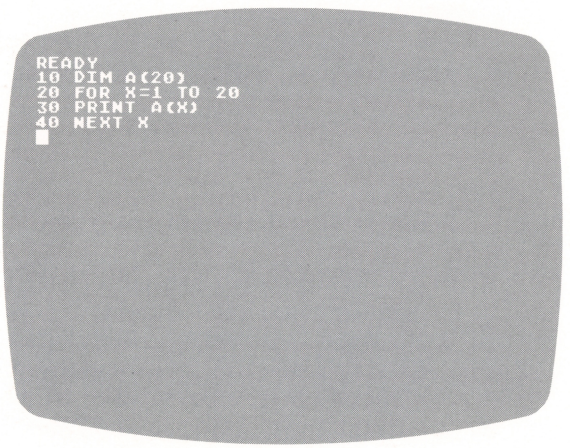
4. The **LET** instruction in the above program, Statement 30, places a value in the array variable. When **READ/DATA** statements and **INPUT** statements are used to place values into array variables, the computer must use these instructions indirectly. For example,

   READ N
   LET A(X)=N, etc.

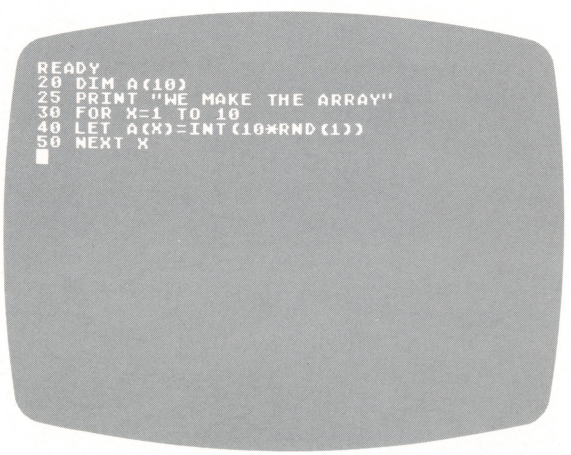
Frame 7: Practice Session for Lesson Three

Note: To clear RAM computer memory,type **NEW** and press RETURN before *every* practice session and between each example program.

Run the following program:



When setting up an array it is necessary to store values in each variable in the array, as in the following example. Run the next program, and then check what is in each variable with a **PRINT** instruction, such as **PRINT A(3)**.



Add the following lines to the previous program and run it again:

```
READY
60 PRINT
70 PRINT "HERE ARE THE VALUES"
80 FOR X=1 TO 10
90 PRINT "A(";X;") = ";A(X)
100 NEXT X
```

Run this sample program:

```
READY
2 GRAPHICS 0
5 PRINT
10 PRINT "JOGGING CALENDER"
20 PRINT "(SHORTENED TO 7 DAYS)"
25 PRINT
30 DIM DISTANCE(7)
40 FOR DAY=1 TO 7
50 PRINT "HOW MANY MILES DID YOU JOG O
N DAY ";DAY
60 INPUT MILES
65 DISTANCE(DAY)=MILES
70 NEXT DAY
80 PRINT
90 PRINT
100 PRINT "RESULTS FOR WHAT DAY?"
110 INPUT DAY
120 PRINT "ON THAT DAY YOU WENT ";DIST
ANCE(DAY);" MILES"
130 GOTO 80
```

Run this sample program:

```
READY
5 GRAPHICS 0
6 DIM FLIGHT(7)
10 PRINT
15 PRINT "THIS IS THE RESERVATION SYST
EM FOR"
20 PRINT "FLY-BY-NIGHT AIRWAYS."
30 PRINT "YOU CAN RESERVE FLIGHTS 1-7."
50 PRINT
60 FOR N=1 TO 7
70 LET FLIGHT(N)=4
80 NEXT N
90 PRINT "WHAT FLIGHT (1 THRU 7) DO YO
U WANT?"
100 INPUT N
110 IF FLIGHT(N)=0 THEN PRINT "SORRY.
ALL FLIGHTS TAKEN.  TRY ANOTHER."
120 IF FLIGHT(N)<>0 THEN FLIGHT(N)=FLI
GHT(N)-1
160 IF FLIGHT(N)<>0 THEN PRINT "ISSUE
TICKET FOR FLIGHT ";N
170 GOTO 90
```

When you are ready, load Lesson Four on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON FOUR: PEEK and POKE

Frame 8: Computer Memory

The computer's raw memory is like a set of post office boxes, where each box contains just one number ranging from 0 to 255. Each box or location is given an address.

An 8K computer has about 8000 different memory locations that can be used for program storage.

Each number in memory can serve to represent a number, a letter, or a computer instruction. For example, the number 76 could stand for the letter **L**, or it could act like a **GOTO** instruction for the computer's machine language. What the number represents exactly depends on where it is in memory.

Frame 9: POKE

POKE 54018, 52	Memory address is 54018 Instruction to turn on Program Recorder is 52.
POKE 54018, 60	Instruction to turn off Program Recorder is 60.

Frame 10: PEEK

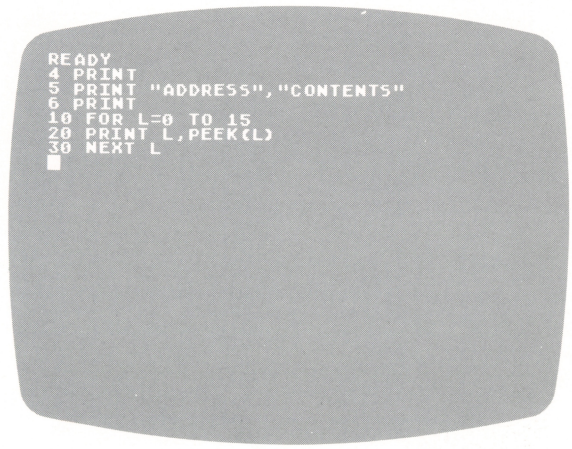
PRINT PEEK (800) prints the number stored in memory location 800.

X=PEEK(800) places the number stored in memory location 800 into the variable X.

Frame 11: Practice Session for Lesson Four

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

Try this program to see the numbers stored in different memory locations:



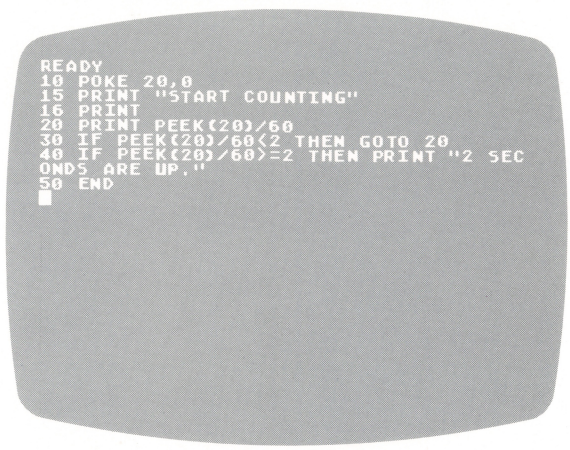
Use **POKE** instructions with caution. No damage results to the computer physically, but you can crash your program! Try this **POKE** to see a small example of what we mean:

POKE 755,4

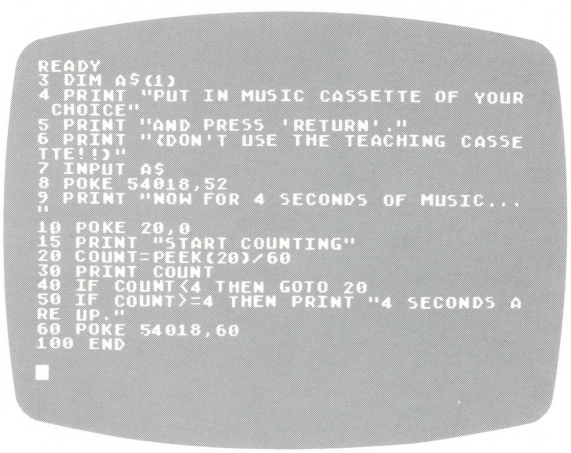
To correct the "damage," type in **POKE 755,2**.

Memory location 20 is used by the computer to count the number of frames the computer puts on your television set. The number in 20 starts at 0 and 1 is added every 1/60th of a second. When the number hits 255, the contents start again at 0.

You can POKE 20 and put the number 0 in it to start. It will then count 1/60ths of a second, as in the following program:



The next program turns on your Program Recorder and allows it to run for 4 seconds. The Program Recorder turns off automatically.



When you are ready, load Lesson Five on the cassette. You must turn over the cassette, Writing Programs Two, to Side 2. Refer to General Instructions for Loading the Cassette Tape. Remember to reset the Program Recorder counter to 000.

LESSON FIVE: ATASCII CODE AND INSTRUCTIONS

Frame 12: Some ATASCII Characters and Their Code Numbers

ATASCII CODE NUMBER	CHARACTER	ATASCII CODE NUMBER	CHARACTER
32	(space)	81	Q
33	!	82	R
34	"	83	S
35	#	84	T
36	$	85	U
37	%	86	V
38	&	87	W
39	'	88	X
40	(	89	Y
41	)	90	Z
42	*	91	[
43	+	92	\
44	,	93	]
45	-	94	^
46	.	95	—
47	/	96	♦
48	Ø	97	a
49	1	98	b
50	2	99	c
51	3	100	d
52	4	101	e
53	5	102	f
54	6	103	g
55	7	104	h
56	8	105	i
57	9	106	j
58	:	107	k
59	;	108	l
60	<	109	m
61	=	110	n
62	>	111	o
63	?	112	p
64	@	113	q
65	A	114	r
66	B	115	s
67	C	116	t
68	D	117	u
69	E	118	v
70	F	119	w
71	G	120	x
72	H	121	y
73	I	122	z
74	J	123	♣
75	K	124	|
76	L	125	(clear screen)
77	M		
78	N		
79	O		
80	P		

(This is a sample of the 255 possible characters of ATASCII)

Frame 13: Interesting ATASCII Characters

PRINT CHR$(125)	Clears television screen
PRINT CHR$(253)	Produces a buzzing sound
PRINT CHR$(28)	Moves cursor up one line (without erasing anything)
PRINT CHR$(29)	Moves cursor down one line
PRINT CHR$(30)	Moves cursor over one space to the left
PRINT CHR$(31)	Moves cursor over one space to the right

Frame 14: Summary of ASC and CHR$

CHR$(N) will produce the character of the ATASCII number.

ASC(L$) will produce the ATASCII number of the character stored in L$.

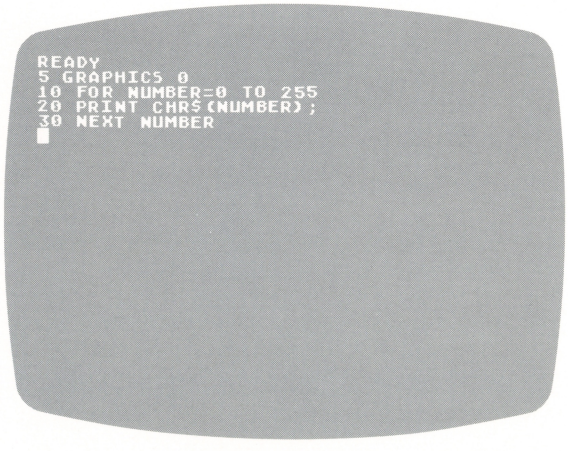
PRINT CHR$(65) will print the letter A from the ATASCII code number 65.

PRINT ASC("A") will print the ATASCII number 65.

Frame 15: Practice Session for Lesson Five

Note: To clear RAM computer memory,type **NEW** and press RETURN before *every* practice session and between each example program.

The following program will print the whole range of ATASCII characters. Note what happens when CHR$(125) is printed.



Add the following lines to keep the screen from being erased.

```
READY
6 PRINT "ATASCII CHARACTERS"

20 IF NUMBER<>125 THEN PRINT CHR$(NUMB
ER);

```

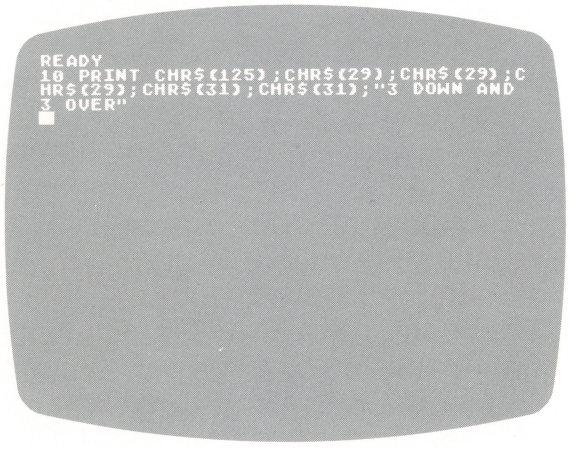
This program will show you the ATASCII code for certain letters and symbols. (Don't use any "invisible" characters.)

```
READY
5 GRAPHICS 0
10 DIM LETTER$(1)
20 PRINT "TYPE IN CHARACTER OR SYMBOL"
;
30 INPUT LETTER$
40 PRINT "THE ATASCII CODE FOR ";LETTER$
;" IS ";
50 PRINT ASC(LETTER$)
60 GOTO 20
```

Press the BREAK key to end program execution. Add this line to the above program and run it again.

```
READY
35 PRINT CHR$(253);
```

Run this line and see what happens. Then try your own program that will print characters on different parts of the television screen.



When you are ready, load Lesson Six on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON SIX: STRING HANDLING

Frame 16

The following instruction will tell the computer to look at the string stored in WORD$ starting with the fifth character:

WORD$(5)

We can print or store the result in another string variable:

PRINT WORD$(5) PART$ = WORD$(5)

Any number can be used in the parentheses just so long as it does not exceed the length of the string stored in the variable.

Example:

STR$ = "A SHORT PHRASE"

P$ = STR$(8)

PRINT P$

PHRASE

Frame 17

The following statement will tell the computer to look at the string stored in WORD$ between two numbers:

WORD$(5,8)

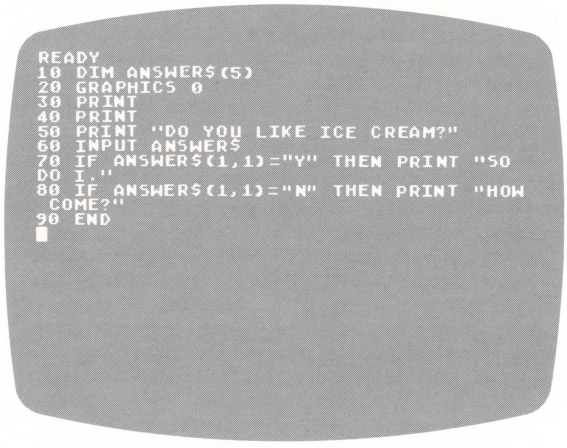
PRINT WORD$(5,8) will now print the string from the fifth to the eighth character.

Example:

```
STR$ = "A SHORT PHRASE"
P$ = STR$(3,7)
PRINT P$
SHORT
```

Frame 18

The following program demonstrates how to get one character from a string that has been input into a string variable. In this case the program looks at the first character only.



Frame 19: Practice Session for Lesson Six

Note: To clear RAM computer memory,type **NEW** and press RETURN before *every* practice session and between each example program.

Enter the following instructions as direct commands:

```
DIM WORD$(40)
LET WORD$ = "A SHORT PHRASE"
```

Now experiment with printing out pieces of the words in WORD$ by using the following kinds of instructions:

PRINT WORD$(3) or PRINT WORD$(5,6) etc.

After trying this a few times, run the following program:

```
READY
5 DIM WORD$(40)
10 GRAPHICS 0
20 PRINT
30 PRINT "TYPE IN WORD"
40 INPUT WORD$
50 PRINT WORD$;" CONTAINS ";LEN(WORD$)
;" LETTERS."
60 PRINT
100 GOTO 20
```

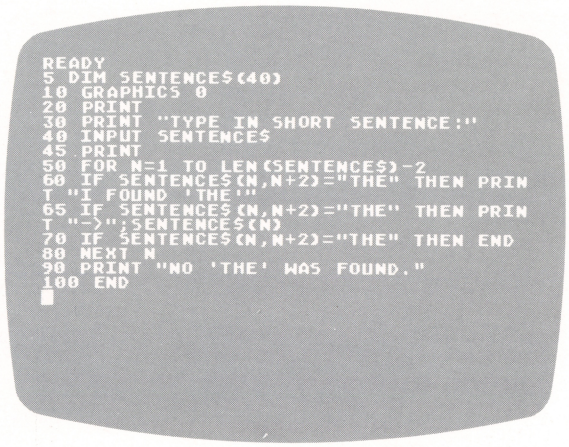
Now add the following lines. (Watch the quotation marks and the semicolons!)

```
READY
70 FOR N=1 TO LEN(WORD$)
80 PRINT "WORD$(";N;")=";WORD$(N)
90 NEXT N
95 PRINT
```

Try this program:

```
READY
5 DIM WORD$(40)
10 GRAPHICS 0
20 PRINT
30 PRINT
40 PRINT "TYPE IN WORD:"
50 INPUT WORD$
60 PRINT "HERE IS THE WORD VERTICALLY-
"
65 PRINT
70 N=0
80 N=N+1
90 PRINT WORD$(N,N)
100 IF N=LEN(WORD$) THEN END
110 GOTO 80
```

The next program is an analysis program. It looks at a sentence for specific words:



When you are ready, load Lesson Seven on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON SEVEN: MULTIPLE STATEMENTS

Frame 20

Separate statements can be placed on the same line in a BASIC program if they are separated with a colon (:).

```
10 GRAPHICS 0
20 PRINT
30 PRINT "THIS IS A VERY SHORT PROGRAM"
40 END
```

The above program can fit all on one line—

```
10 GRAPHICS 0: PRINT : PRINT "THIS IS A VERY SHORT PROGRAM": END
```

Frame 21

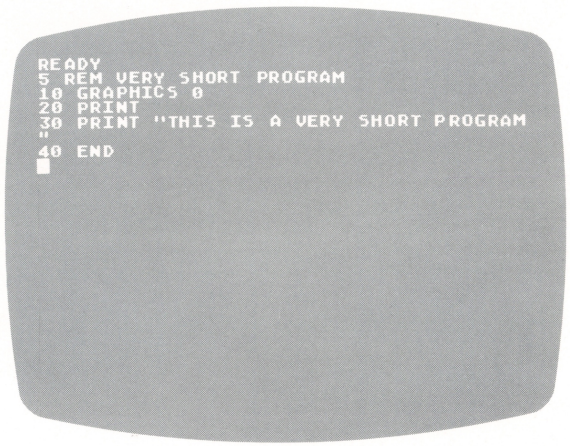
The **IF/THEN** condition determines whether statements that follow on the same line will be executed.

```
10 INPUT A
20 IF A=0 THEN PRINT "TRY AGAIN": GOTO 10
30 PRINT "THAT'S ALL": END
```

In the above program, the computer will reach the GOTO 10 statement only when A=0. If A<>0, then the computer ignores everything after the **IF/THEN** statement on the same line.

Frame 22: REM (or Remark) Statement

The **REM** statement is a label only. The computer will ignore it. Here the **REM** statement has been added to the program used previously:



The computer will also ignore any statement that follows a **REM** statement on the same line.

```
10 REM THE NEXT STATEMENT WILL BE IGNORED: GRAPHICS 0
```

Be very careful with the following in multistatement lines:

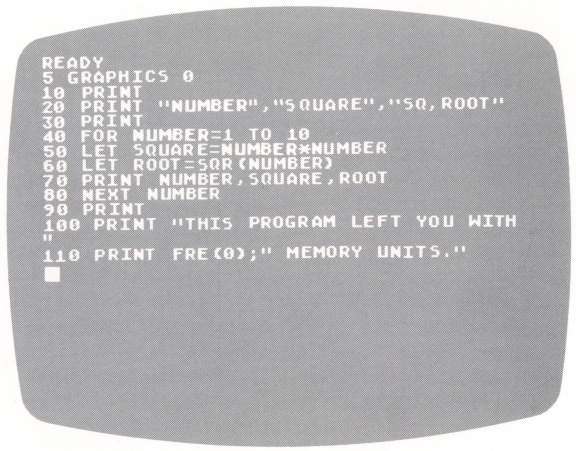
REM
DATA
IF/THEN
GOTO

It is safest to have each **DATA** statement on its own line with no other statements.

Frame 23: Practice Session for Lesson Seven

Note: To clear RAM computer memory, type **NEW** and press RETURN before EVERY practice session and between each example program.

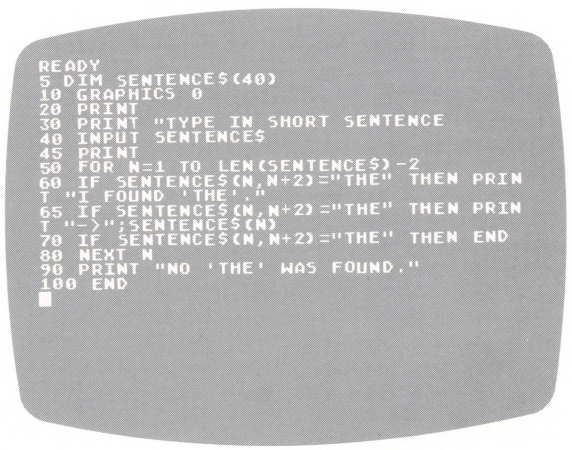
The following program appeared in an earlier lesson. We have added line 110 to the program so that you can see how much memory the program used. Run the program, and check how much memory you have left.



```
READY
5 GRAPHICS 0
10 PRINT
20 PRINT "NUMBER","SQUARE","SQ.ROOT"
30 PRINT
40 FOR NUMBER=1 TO 10
50 LET SQUARE=NUMBER*NUMBER
60 LET ROOT=SQR(NUMBER)
70 PRINT NUMBER,SQUARE,ROOT
80 NEXT NUMBER
90 PRINT
100 PRINT "THIS PROGRAM LEFT YOU WITH
"
110 PRINT FRE(0);" MEMORY UNITS."
```

Now rewrite the above program using multiple statement lines, and see how much memory space the new version occupies.

Rewrite the program below using multiple statement lines. See if you can get rid of two of the **IF/THEN** statements.



```
READY
5 DIM SENTENCE$(40)
10 GRAPHICS 0
20 PRINT
30 PRINT "TYPE IN SHORT SENTENCE
40 INPUT SENTENCE$
45 PRINT
50 FOR N=1 TO LEN(SENTENCE$)-2
60 IF SENTENCE$(N,N+2)="THE" THEN PRIN
T "I FOUND 'THE'."
65 IF SENTENCE$(N,N+2)="THE" THEN PRIN
T "->";SENTENCE$(N)
70 IF SENTENCE$(N,N+2)="THE" THEN END
80 NEXT N
90 PRINT "NO 'THE' WAS FOUND."
100 END
```

When you are ready, load Lesson Eight on the cassette by typing **CLOAD** and pressing RETURN twice. After the READY message appears on the screen, type **RUN** and press RETURN to start the lesson.

LESSON EIGHT: SUBROUTINES

Frame 24: Purpose of a Subroutine

In this program, we use the same routine several times, so we moved it to the end. Now it's called a subroutine.

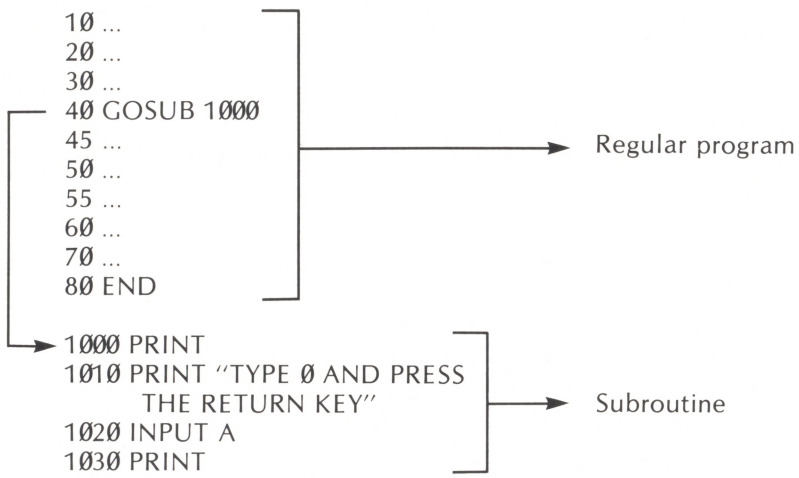
```
1Ø ...
2Ø ...
3Ø IF A=1 THEN PRINT "YES"
4Ø ...
5Ø ...
6Ø IF A=1 THEN PRINT "YES"
7Ø ...
8Ø END

1ØØØ IF A=1 THEN PRINT "YES"
```

Routines taken out of program.

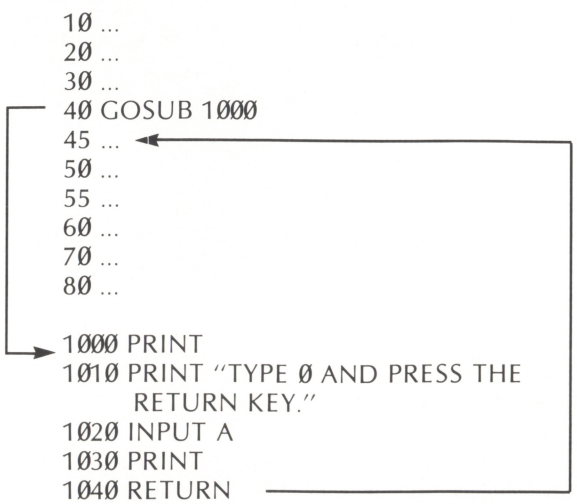
Subroutine written just once.

Frame 25: GOSUB and **RETURN** Statements

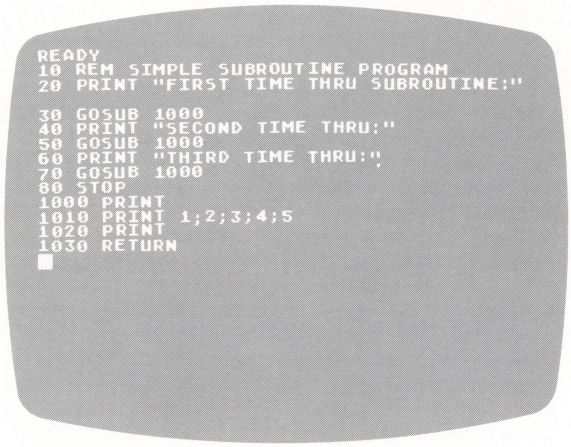
When the computer reaches the **GOSUB** statement in Line 40, it goes to Line 1000.



Here is how the **RETURN** statement works:



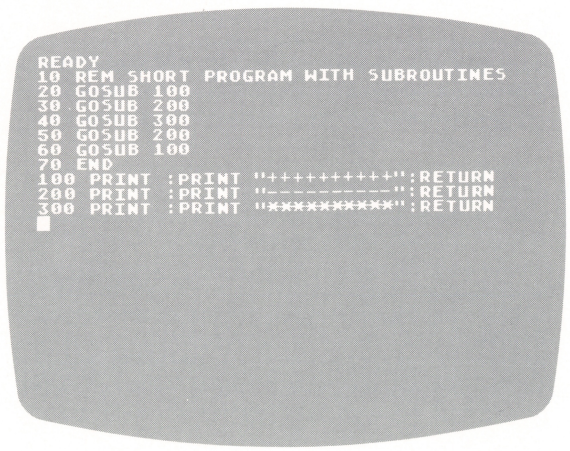
Frame 26: Question Example



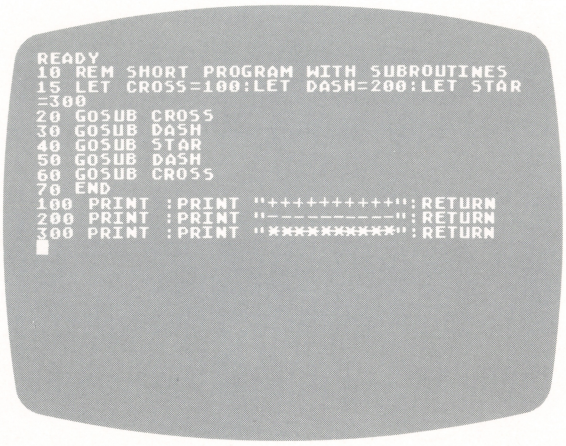
Frame 27: Practice Session for Lesson Eight

Note: To clear RAM computer memory, type **NEW** and press RETURN before *every* practice session and between each example program.

The following program has three subroutines. Run it and see what happens.

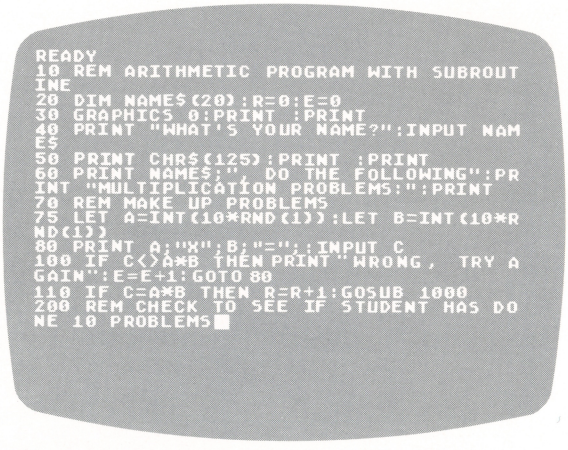


The following frame is another version of the program above. However, the **GOSUB** statements use variables instead of numbers.

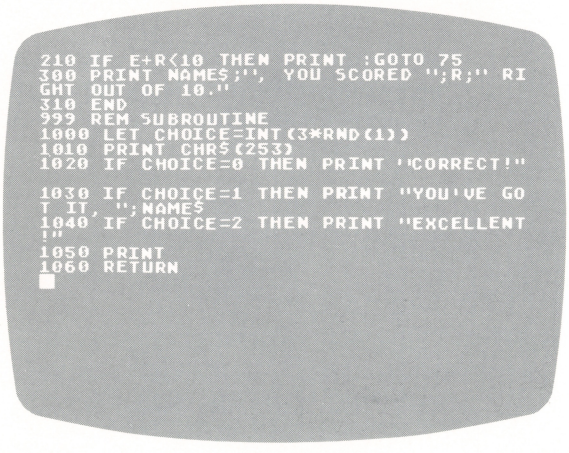


```
READY
10 REM SHORT PROGRAM WITH SUBROUTINES
15 LET CROSS=100:LET DASH=200:LET STAR
=300
20 GOSUB CROSS
30 GOSUB DASH
40 GOSUB STAR
50 GOSUB DASH
60 GOSUB CROSS
70 END
100 PRINT :PRINT "++++++++++":RETURN
200 PRINT :PRINT "----------":RETURN
300 PRINT :PRINT "**********":RETURN
```

The following program illustrates many of the points taught in the last two lessons.



```
READY
10 REM ARITHMETIC PROGRAM WITH SUBROUT
INE
20 DIM NAME$(20):R=0:E=0
30 GRAPHICS 0:PRINT :PRINT
40 PRINT "WHAT'S YOUR NAME?":INPUT NAM
E$
50 PRINT CHR$(125):PRINT :PRINT
60 PRINT NAME$;", DO THE FOLLOWING":PR
INT "MULTIPLICATION PROBLEMS:":PRINT
70 REM MAKE UP PROBLEMS
75 LET A=INT(10*RND(1)):LET B=INT(10*R
ND(1))
80 PRINT A;"X";B;"=";:INPUT C
100 IF C<>A*B THEN PRINT " WRONG, TRY A
GAIN":E=E+1:GOTO 80
110 IF C=A*B THEN R=R+1:GOSUB 1000
200 REM CHECK TO SEE IF STUDENT HAS DO
NE 10 PROBLEMS
```

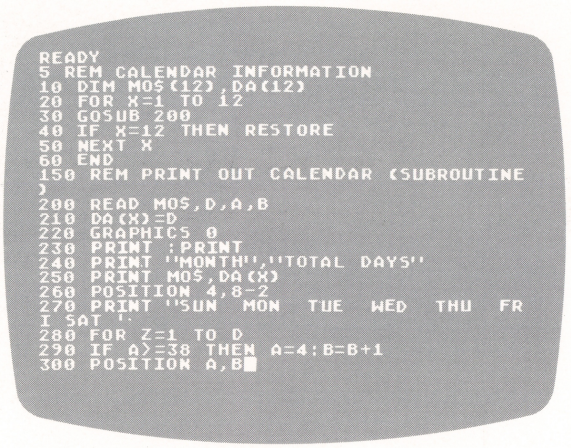


```
210 IF E+R<10 THEN PRINT :GOTO 75
300 PRINT NAME$;", YOU SCORED ";R;" RI
GHT OUT OF 10."
310 END
999 REM SUBROUTINE
1000 LET CHOICE=INT(3*RND(1))
1010 PRINT CHR$(253)
1020 IF CHOICE=0 THEN PRINT "CORRECT!"
1030 IF CHOICE=1 THEN PRINT "YOU'VE GO
T IT, ";NAME$
1040 IF CHOICE=2 THEN PRINT "EXCELLENT
!"
1050 PRINT
1060 RETURN
```

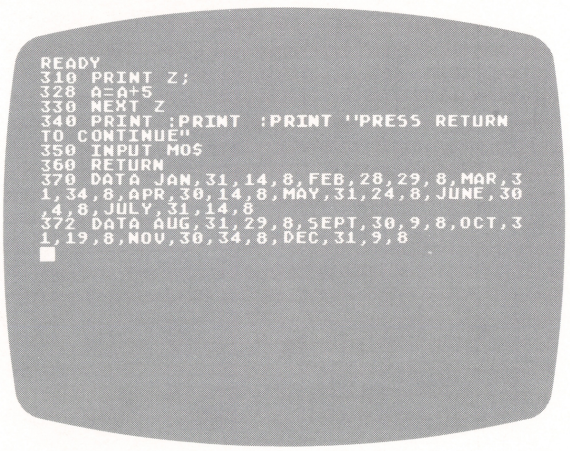
CONGRATULATIONS! GO ON TO THE FINAL QUIZ ON THE CASSETTE.

You should be ready now for the Final Quiz for Writing Programs Two. Your score will automatically appear on the screen at the conclusion of the quiz. To load the Final Quiz, use the same procedure you used to load the other lessons.

Frame 28: Program to Use With Final Quiz



```
READY
5 REM CALENDAR INFORMATION
10 DIM MO$(12),DA(12)
20 FOR X=1 TO 12
30 GOSUB 200
40 IF X=12 THEN RESTORE
50 NEXT X
60 END
150 REM PRINT OUT CALENDAR (SUBROUTINE
)
200 READ MO$,D,A,B
210 DA(X)=D
220 GRAPHIC$ 0
230 PRINT :PRINT
240 PRINT "MONTH","TOTAL DAYS"
250 PRINT MO$,DA(X)
260 POSITION 4,8-2
270 PRINT "SUN  MON  TUE  WED  THU  FR
I  SAT ".
280 FOR Z=1 TO D
290 IF A>=38 THEN A=4:B=B+1
300 POSITION A,B
```



```
READY
310 PRINT Z;
320 A=A+5
330 NEXT Z
340 PRINT :PRINT :PRINT "PRESS RETURN
TO CONTINUE"
350 INPUT MO$
360 RETURN
370 DATA JAN,31,14,8,FEB,28,29,8,MAR,3
1,34,8,APR,30,14,8,MAY,31,24,8,JUNE,30
,4,8,JULY,31,14,8
372 DATA AUG,31,29,8,SEPT,30,9,8,OCT,3
1,19,8,NOV,30,34,8,DEC,31,9,8
```

Now that you have successfully completed this course, you have acquired the necessary skills for **AN INVITATION TO PROGRAMMING™ 3**. Course content includes BASIC programming of sound and graphics.

LIMITED 90 DAY WARRANTY
ON ATARI® PERSONAL COMPUTER PRODUCTS

ATARI, INC. ("ATARI") warrants to the original consumer purchaser that this ATARI Personal Computer Product (not including computer programs) shall be free from any defects in material or workmanship for a period of 90 days from the date of purchase. If any such defect is discovered within the warranty period, ATARI's sole obligation will be to repair or replace, at its election, the Computer Product free of charge on receipt of the unit (charges prepaid, if mailed or shipped) with proof of date of purchase satisfactory to ATARI at any authorized ATARI Service Center. For the location of an authorized ATARI Service Center nearest you, call toll-free:

In California (800) 672-1430
Continental U.S. (800) 538-8547

or write to:

Atari, Inc.
Customer Service Department
1340 Bordeaux Drive
Sunnyvale, CA 94086

YOU MUST RETURN DEFECTIVE COMPUTER PRODUCTS TO AN AUTHORIZED ATARI SERVICE CENTER FOR IN-WARRANTY REPAIR.

This warranty shall not apply if the Computer Product: (i) has been misused or shows signs of excessive wear, (ii) has been damaged by being used with any products not supplied by ATARI, or (iii) has been damaged by being serviced or modified by anyone other than an authorized ATARI Service Center.

ANY APPLICABLE IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE HEREBY LIMITED TO NINETY DAYS FROM THE DATE OF PURCHASE. CONSEQUENTIAL OR INCIDENTAL DAMAGES RESULTING FROM A BREACH OF ANY APPLICABLE EXPRESS OR IMPLIED WARRANTIES ARE HEREBY EXCLUDED. Some states do not allow limitations on how long an implied warranty lasts or do not allow the exclusion or limitation of incidental or consequential damages, so the above limitations or exclusions may not apply to you.

This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

DISCLAIMER OF WARRANTY
ON ATARI COMPUTER PROGRAMS

All ATARI computer programs are distributed on an "as is" basis without warranty of any kind. The entire risk as to the quality and performance of such programs is with the purchaser. Should the programs prove defective following their purchase, the purchaser and not the manufacturer, distributor, or retailer assumes the entire cost of all necessary servicing or repair.

ATARI shall have no liability or responsibility to a purchaser, customer, or any other person or entity with respect to any liability, loss, or damage caused directly or indirectly by computer programs sold by ATARI. This disclaimer includes but is not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer programs.

REPAIR SERVICE

If your ATARI Personal Computer Product requires repair other than under warranty, please contact your local authorized ATARI Service Center for repair information.

IMPORTANT: If you ship your ATARI Personal Computer Product, package it securely and ship it, charges prepaid and insured, by parcel post or United Parcel Service.

ATARI®

PRINTED IN U.S.A.

A Warner Communications Company

C015727-06 REV. 1