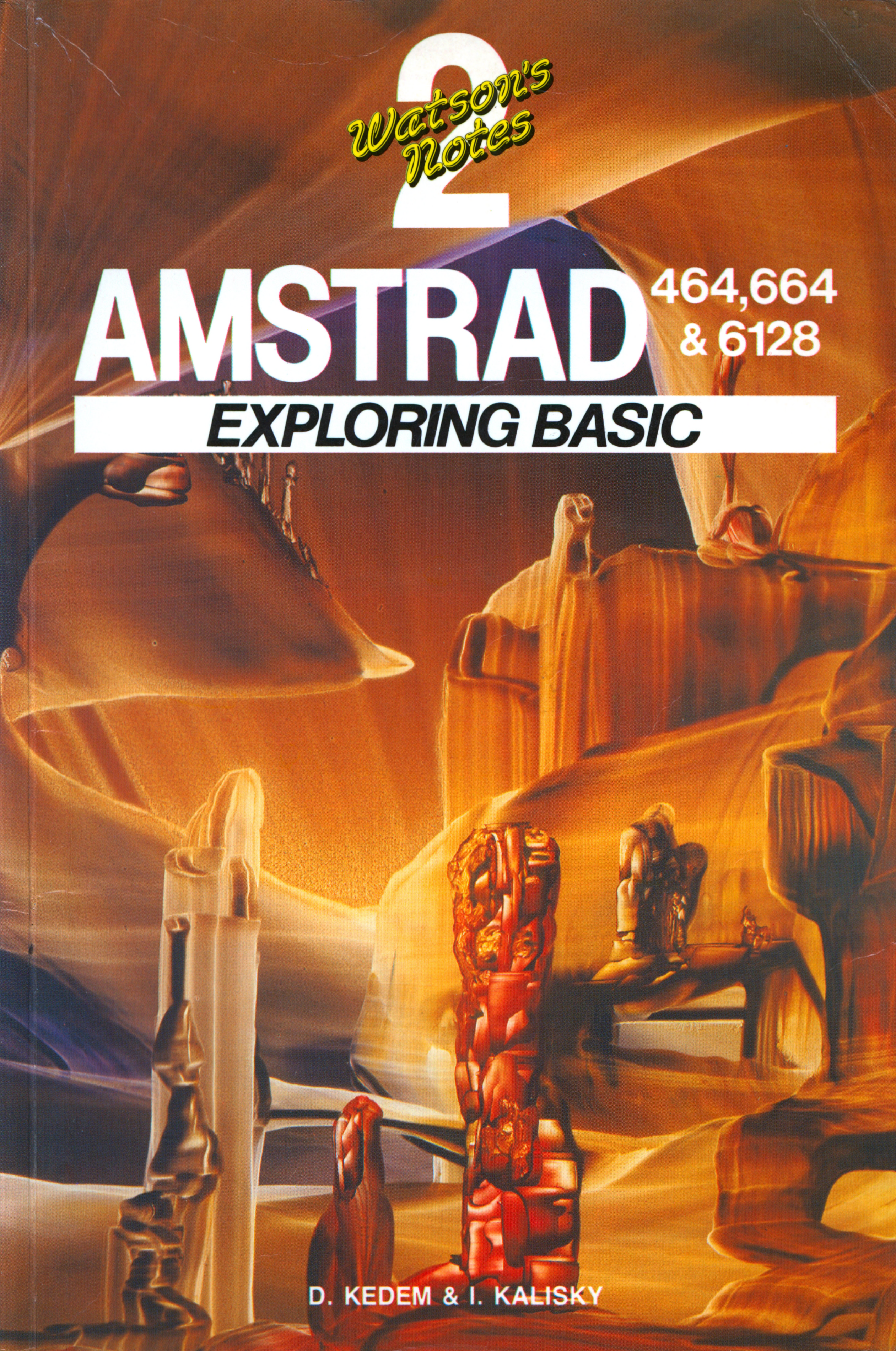
Watson's Notes 2

AMSTRAD 464,664 & 6128

EXPLORING BASIC



D. KEDEM & I. KALISKY

Watson's Notes

AMSTRAD 464, 664 & 6128

I.KALISKY & D.KEDEM

UNIT 2

Exploring BASIC

GLENTOP publishers

NOVEMBER 1985

ISBN 1 85181 002 1

Published by:

Glentop Publishers Ltd
Standfast House
Bath Place
High Street
Barnet
Herts EN5 1ED
Tel: Ø1-441-413Ø

Cover illustration from an original painting by Nick Woods.

CONTENTS

FOREWORD

We will start off this unit by demonstrating the Amstrad's ability to produce colour and sound.

Type the following program into the computer and run it. (At this stage, don't worry if you don't understand the program.)

```
10 FOR N=1 TO 20
20 INK 0,N
30 BORDER 20-N
40 SOUND 1,5*N,100
50 NEXT N
60 GOTO 10
```

In the course of this unit, you will learn all the new commands and techniques that we have incorporated in this program, such as FOR...NEXT loops, Numeric Variables and interesting screen displays. That is to say, printing characters on the screen, constructing shapes out of characters and moving individual characters around the screen. However, it is the commands and techniques you will use during this unit that will be of the greatest importance to you in learning how to program your computer.

Itzhak Kalisky and Dani Kedem.

Chapter 1

NUMERIC VARIABLES

Look at the program in the Foreword. You can see that 'N' appears in almost every line.

What does this mysterious letter N do? We will show you throughout this chapter.

To begin with, erase the program in the computer's memory.

(If you don't remember how, turn to Unit 1, Chapter 3.)

Type the following program into the computer exactly as it is written:

```
10 MODE 0
20 LET N=1
30 LOCATE 10,10:PRINT N
```

When you run the program, the number ________(1, 10) appears on the screen.

What should be done to line 20 so that the computer prints 2 in the centre of the screen?

Task

Alter the program so that the computer prints the numbers from 1 onwards, in the centre of the screen. (Each number appears in the same position.)

In order to do this, you must know what a VARIABLE is, and how to use it.

Instruct the computer to:

```
LET A=6
```

and

```
LET B=4
```

Don't forget to press ENTER after each command.

What will happen now, if you tell the computer to:

```
PRINT A+B
```

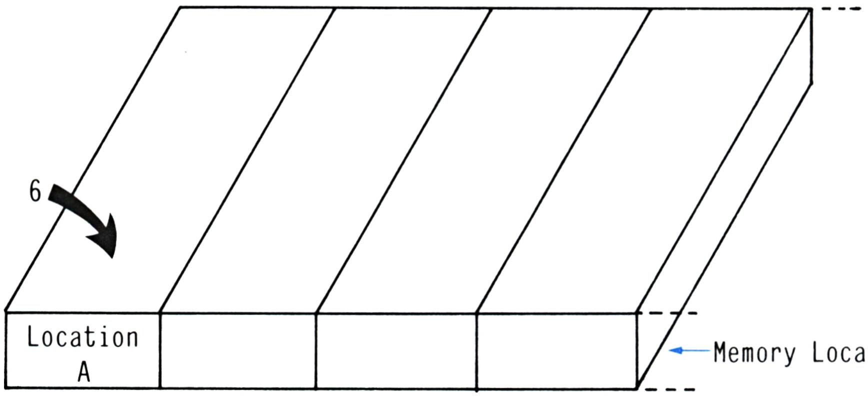
Think about it before trying it.

The computer prints the number ___10___ (Ø, 6, 4, 1Ø). Why does this happen?

When you typed in LET A=6, the computer did the following:

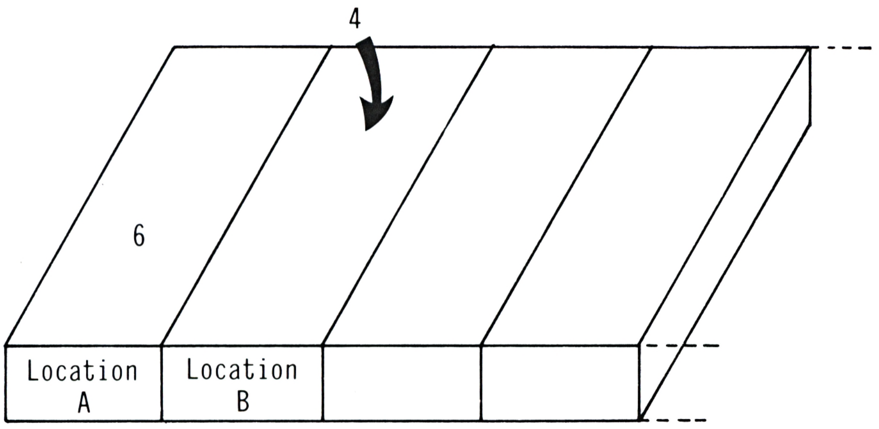
It put the number 6 into one of its many MEMORY LOCATIONS and called the location A.

Memory Locations



A similar thing happened when you executed (ENTERed) LET B=4.

It put the number 4 into another memory location and called the location B.



When you came to PRINT A+B, the computer did the following:

- It looked for the first location (A) and copied the number found in it (6) into the computers calculating unit.

- The computer then looked for the second location (B) and copied the number in it (4) into the same calculating unit.

This calculating unit is referred to as the ARITHMETIC AND LOGIC UNIT or ALU for short.

Arithmetic and Logic Unit

It is in the ALU that the computer adds the two numbers (6 and 4). The result is then printed.

? Do the numbers 6 and 4 still exist in locations A and B after this addition? YES

Tell the computer to print the value of A (PRINT A).

? And what about B? Does its value remain the same?

Note

It is important to note that when an arithmetic operation is performed, the values of the locations are only 'copied' and not 'taken'. Therefore, the values remain in their respective locations.

? What does the computer display after executing PRINT A*B, remembering that the '*' is the multiplication symbol?

☐ Check your answer on the computer.

☐ Now tell the computer to do the following:

```
LET A=15
```

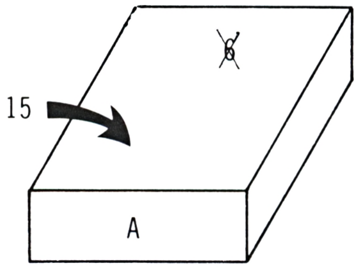
? Now what will the computer display after telling it to PRINT A; 15 (5, 6, 15 or 15 and 6)?

☐ Check your answer. Were you correct? yes

Explanation

- When you executed LET A=6, the computer put 6 into memory location A.

- When you enter LET A=15, the computer searches for memory location A and puts the number 15 in it. ie. 6 is erased.



Conclusion

We have seen how memory location A can represent a number and how that number can be changed.

Variables

For this reason, A is referred to as a VARIABLE, or more precisely, a Numeric Variable. (Later on in this unit you will meet another variable called a String Variable.)

Set the variable B to the value of 5.

What will the computer print when you tell it to:

```
PRINT A/B

PRINT 3*A
```

What will the computer do when you instruct it to:

```
LET C=A+B
```

In this instance, we have defined a new variable, C, which is equal to the sum of the two variables A and B. (Check this by printing C.)

What will the computer print in the following case:

```
PRINT D
```

Explanation

The computer checks if a variable called D exists in memory and if it doesn't, D receives a value of Ø.

What will happen when you type in:

```
PRINT A/F
```

Why does the computer inform you that it has divided by zero?

Answer 1

Shortening the LET command

Up until now, we have defined variables with the LET command, however, the Amstrad computer knows how to do this without using LET.

Type in the following:

```
A=5Ø
```

Now tell the computer to print the value of A.

Variable Names

Variable Names

So far, the variable names we have used have consisted of 1 letter eg. A and B, however, as long as the name starts with a letter, it can be followed by numbers or more letters.

From the rules above, which of the following are illegal variable names?

```
AMSTRAD, SUM, B, 2D

A1B2, 25, PRINT
```

Answer 2

Type the following into the computer:

```
1X=2
```

Now print the value of the variable (PRINT 1X).

The computer replies by printing:

```
    1  Ø
```

Now tell the computer to LIST the program.

Why is there an extra line in the program?

The reason for this is that the computer accepted the '1' at the start of the line as a line number. Therefore, when you printed '1X', the computer displayed the number 1 and next to it, the value of variable X which equals Ø.

Delete line 1.

(If you don't remember how to, turn to Unit 1, Chapter 3.)

Conclusion

- A variable name must begin with a letter.
- Either numbers or letters may follow the first letter.

Challenge

Define a new variable, X, equal to 1Ø:

```
    X=1Ø
```

What will be the value of X after executing (entering) the following command:

```
    LET X=X+1
```

In order to find this out, type in the command above (this can also be used without LET) and then print the value of X.

X should be equal to ________(1, 1Ø, 11)?

Why was this result produced?

To understand this, we must look at what the computer does.

- The computer firstly acts on the right-hand side of the equation. In our case, it takes the current value of X and adds 1 to it (1Ø+1).

- The computer then puts the value obtained from the right-hand side of the equation into the variable specified on the left-hand side. In our case, the variable is X again so X now receives the value of 11.

Note: the variable on the left of the equation can be any variable.

Task

Now, after understanding variables, you can work on the task we set you earlier.

Display the program that is already in memory. (This is the program on Page 7.)

Alter the program so that the computer prints the numbers from 1 onwards in the centre of the screen. ie. print 1 in the centre of the screen and afterwards, at the same position, print 2, and then 3 etc.

You should be able to write this program on your own, but if you have any difficulties, there are some hints on the next page.

Hint 1: two lines should be added to the program.

Hint 2: complete the following lines:

```
40 _____ = _____+1
50 GOTO _____
```

If the numbers are displayed too quickly:

- Stop the program running by pressing the ESC key twice and insert a 'time-delay' in the appropriate place in the program similar to that in Roberts Exercises (Unit 1, Chapter 7).

Answer 3

Does the result of this program remind you of a digital watch? In Chapter 3, we will write a program that constructs a watch like this.

We will now explain tó you how the program works:

Line 20: After executing this line, N is equal to __________(1, 2, 3).

Line 30: At this point, the computer prints the value of N.

Line 40: N receives a new value.

```
  N=N+1
  ↑ ↑
New  Current
```

When carrying out this line for the first time, N receives a value of __________(1, 2, 3).

Line 50: This command is already familiar to you (Unit 1). It causes the program to go to line __________

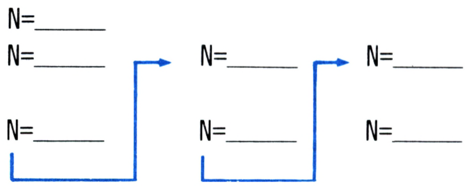
A Dry Run

Dry Runs

A DRY RUN is a testing technique whereby you, the programmer, act as the computer by going through the program line by line and writing down the various values of the variables during a run.

Therefore, in order to follow the changes that N undergoes in our program, do a dry run.

```
1Ø MODE Ø
2Ø N=1
3Ø LOCATE 1Ø,1Ø:PRINT N
35 FOR T=1 TO 3ØØ:NEXT T
4Ø N=N+1
5Ø GOTO 3Ø
```



Check your result with Answer 4.

Answer 4

Whatever you do, don't wait for the program to stop running or you'll be sitting there for ages.

In fact, the computer will count up to 1E+38, which means, 1 followed by 38 zeros.

Task

Make the following changes to the program in memory:

- Only the odd numbers are printed:

  1,3,5,7...

Answer 5

- Only numbers that can be exactly divided by 5 (multiples) are printed:

  5,1Ø,15...

What will happen if you change line 5Ø to:

```
5Ø GOTO 2Ø
```

Games Corner

☐ Erase everything in the computers memory (NEW) and type in and run the following program:

```
10 LOCATE 10,10:PRINT "BOOM"
20 B=B+1
30 GOTO 20
```

? Is the computer incrementing (adding to) the variable B? ________(Yes or No)?

It is incrementing B, but how do we know if the value of B is not printed.

☐ Stop the computer running and print the value of B. (If you don't remember how to, turn to page15.)

The number printed is the value of B when the program stopped.

☐ Run the program and after stopping it, print the value of B again.

? What is the value of B this time?

A Game

The aim of the game is to stop the computer running as near as possible to 1000 (B=1000).

Rules: each player, in turn, receives 5 attempts. On each attempt, the player runs the program and stops it when he/she thinks B has reached 1000.

Each player remembers their best score (the number nearest to 1000) and whoever is nearest, wins.

When you have finished playing with the program, erase it from memory and continue.

Puzzle

What will happen when you run the following program:

```
10 SOUND 1,P,20
20 P=P+100
30 GOTO 10
```

The computer should have stopped and announced:

```
Improper argument in 10
```

Why has this happened?

Print the value of P.

Explanation

You should already know, from Unit 1, that the second arguement in the SOUND command is the pitch of the sound to be produced. This number can have a maximum value of 4095, which is the deepest note producable on this computer. When P receives the value of 4100, the computer can't execute the command and so it informs you of an error.

What, in your opinion, will happen in this case:

```
5 P=4095
10 SOUND 1,P,20
20 P=P-100
30 GOTO 10
```

What do you think the value of P is when the program stops running?

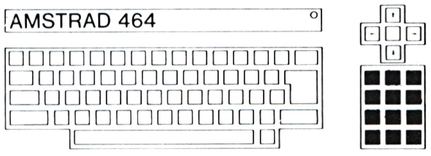
Check your answer on the computer.

Programmable Keys

To end this chapter, we will show you how you can speed up your operation of the Amstrad computer.

Up until now, you have had to type in every letter of a command, eg. LIST. However, the Amstrad computer allows you to shorten this typing process.

You can define the 'number pad', which is located on the right of the keyboard, to store commands. (On the Amstrad 664 and 6128 models, the number pad keys are prefixed with an 'f'.)



Number Pad

For example, if you can't be bothered to type in LIST every time you want to display a program, you can program key 9 or any other key on the number pad to hold the word LIST. (f9 on 664 and 6128 models.)

Every key on the Amstrad has a 'key number'. Below is a list of the key numbers that we will use:

7 135	8 136	9 137
4 132	5 133	6 134
1 129	2 13Ø	3 131
Ø 128		

The diagram above corresponds to the number pad on the computer, where the key number of 9 equals 137. Therefore, if you type in:

```
    KEY 137,"LIST"
```

the computer will hold the word between the quotes (LIST) in the key specified (9) so that whenever you press 9 on the number pad, the word LIST appears instead of the number 9.

Now, providing you haven't erased what was in the computers memory, press ENTER and see if the computer executes the LIST command.

It is also possible to program the keys so that they hold more than one command. For example:

```
KEY 136,"MODE 2:INK 1,4:INK Ø,15:CLS"
```

Type this in and try it out by pressing the appropriate key followed by ENTER. (Refer to the diagram on the previous page for the key to be pressed.

Note

At some point in time, you may want the computer to automatically execute the key you have programmed. For example, if you program RUN into a key, you still have to press ENTER in order to execute the program in memory.

Type in the following command:

```
KEY 135,"RUN"+CHR$(13)
```

Now, if you press the 7 key on the number pad, the program in memory will automatically run. i.e.you don't have to press ENTER.

We won't explain to you in detail what CHR$(13) does, as it will be explained in a later chapter. For now, all you need to know is that CHR$(13) is a special code for ENTER.

Type in NEW.

? Have the programmable keys returned to their original state?

? What about after you press CTRL, SHIFT and ESC simultaneously.

(If you don't remember what happens when you press these three keys simultaneously, turn to Unit 1, Chapter 5.)

We can see, from the result of pressing CTRL, SHIFT and ESC, that when you turn the computer off, you also loose the contents of the programmable keys. (The two operations do the same thing.)

Therefore, we suggest that you program the keys via a 'key definition' program and save the program on either tape or disc depending on which Amstrad you have.

Chapter 2

CONDITIONAL LOOPS

Now you know how to use variables, we will look at another command which incorporates them.

Do you recognise the commands below?

```
10 FOR N=0 TO 15
.
.
.
50 NEXT N
```

They appeared in the program at the start of the Unit, on Page 6.

FOR...NEXT Loops

FOR...NEXT

Do you remember this program from Unit 1:

```
10 PRINT "Your name"
20 GOTO 10
```

The GOTO command in line 20 makes the computer loop around lines 10 and 20 continuously. This sort of loop is called an UNCONDITIONAL LOOP and will only stop when the ESC key has been pressed twice.

Unconditional Loops

With the help of the FOR...NEXT command, you can specify the amount of times you want to carry out a series of commands.

Can you guess how many times the computer will print 'name' on the screen in the following program?

```
10 FOR N=1 TO 5
20 PRINT "name"
30 NEXT N
```

Type the program into the computer and run it.

You can see that 'name' appears 5 times, each one on a different line.

Change the program so that the computer prints your name 8 times.

Answer 6

Conclusion

When you write a program with a FOR...NEXT loop in it, you can control the amount of times the computer executes the lines between the FOR and NEXT commands. For example:

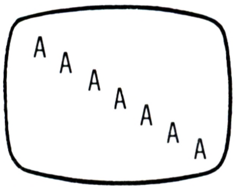
```
 10 FOR N=1 TO 20
┌20 ....
│30 ....
└40 ....
 50 NEXT N
```

How many times will the computer execute lines 20, 30 and 40?

Answer 7

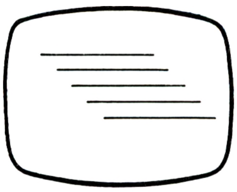
Task

Write a program that will print seven A's diagonally on the screen using a FOR...NEXT loop, in MODE 1.



Answer 8

Using FOR...NEXT, write a program that will print your name on the screen 5 times, like this:



Answer 9

What will this program do?

```
5  X=1
1Ø FOR N=1 TO 2Ø
2Ø PRINT X
3Ø X=X+1
4Ø NEXT N
5Ø PRINT "END"
```

After thinking about it, type it in and run the program. (Remember to erase the previous program.)

As you can see, the program displays the numbers from 1 to 2Ø and prints END when it finishes.

However, it is possible to shorten the program to this:

```
1Ø FOR N=1 TO 2Ø
2Ø PRINT N
3Ø NEXT N
4Ø PRINT "END"
```

Note

The variable N that appears in line 2Ø, also appears in lines 1Ø and 3Ø.

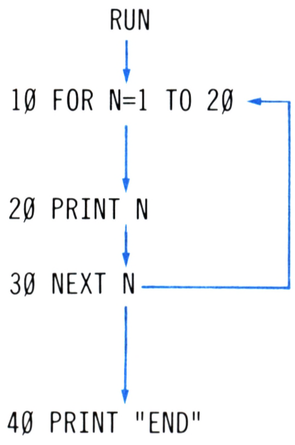
In order to help you understand what happens to N during program execution, we will do a dry run.

Line 1Ø: this line tells the computer to set N equal to 1 and to finish when N equals 2Ø i.e. the loop is performed 2Ø times.

Line 2Ø: this line prints the value of N. ie. the first time around the loop, N=1, the second time, N=2 etc.

Line 3Ø: this line increments (adds 1 to) N and tells the computer to go to line 1Ø.

However, if N is greater than 2Ø ie. N=21, the computer will not go to line 1Ø but instead continue to line 4Ø where it prints END.



```
RUN
1Ø FOR N=1 TO 2Ø
2Ø PRINT N
3Ø NEXT N
4Ø PRINT "END"
```

If N is greater than 2Ø, perform line 4Ø.

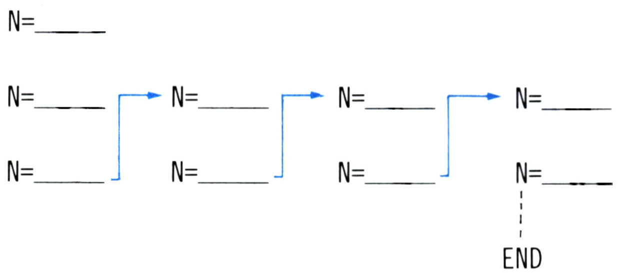
We will now check to see if you have understood the program. Fill in the blank spaces:

```
1Ø FOR N=1 TO 3

2Ø PRINT N

3Ø NEXT N

4Ø PRINT "END"
```



Note: we have made the loop smaller (line 1Ø) to make the test shorter.

Answer 10

We will now return to the program in memory:

What happens if we change line 1Ø to:

```
10 FOR N=5 TO 20
```

You can see that the program prints the numbers from 5 to 20 ?

What happens if you delete line 3Ø? Try it.

The computer announces:

```
NEXT missing in 1Ø
```

The computer produces an error because there is no NEXT command returning the computer to line 2Ø.

Erase the program from memory.

What will the following program do:

```
1Ø FOR N=Ø TO 5
2Ø PRINT N*N
3Ø NEXT N
```

Think about it before checking your answer on the computer.

Delay Loops

Delay Loops

What happens when there are no lines between the FOR and NEXT commands in a program?

```
 1Ø PRINT "START"
┌2Ø FOR T=1 TO 1ØØØ
└3Ø NEXT T
 4Ø PRINT "END"
```

What does this program do and what is the point of lines 2Ø and 3Ø?

After thinking about it, erase the computer's memory and enter the program.

What must you do to double the time period between START and END?

Answer 11

Lines 2Ø and 3Ø produce a delay.

When the computer comes to execute lines 2Ø and 3Ø, it finds a loop and executes it ________(1Ø, 1ØØ, 1ØØØ) times. (The variable N starts at 1 and only when N is greater than 1ØØØ does the computer exit the loop.)

The delay is created by the time the computer takes to complete 1ØØØ loops. Therefore, the more loops the computer makes, the longer the delay.

This method is often used to obtain a time delay and usually written in one line, separated by a colon (:).

$$\texttt{2Ø}\ \underbrace{\texttt{FOR T=1 TO 1ØØØ}}_{1}\texttt{:}\underbrace{\texttt{NEXT T}}_{2}$$

Change the program so that the loop is in line 2Ø, as shown above.

Now run the program and see if the same result is received. (Remember to delete line 3Ø.)

The Tired Program

Look at the following program:

```
10 PRINT X;
20 FOR T=1 TO X*X:NEXT T
30 X=X+1
40 GOTO 10
```

Why have we called this program the 'Tired Program'?

Run the program and see for yourself. (If you don't understand why, look at Answer 12.) **Answer 12**

A Flashing Screen

Write a program that will change the colour of the screen (MODE Ø) from black (Ø) through to orange (15). (If you have a black and white monitor, the different colours will be represented by different shades.)

Add in a delay to slow down the colour change.

When all the colours have been displayed, repeat the whole process using GOTO. **Answer 13**

Now add a line to the program which will change the **border** colour from orange through to black (15 to Ø) at the same time that the screen colour changes. **Answer 14**

What must you add to the program in order to produce a note whose pitch starts off high and gets lower? **Answer 15**

Does this program remind you of anything?

It should do because it does the same as the program at the start of the unit on page 6.

Concluding Challenge

The program will never stop because of the unconditional loop at line 6Ø.

```
60 GOTO 20
```

If we put in another FOR...NEXT loop, we can make the program stop when we want it to.

☐ Complete the following FOR...NEXT commands to make the program run 3 times only.

```
15 FOR T=1 TO ____

60 NEXT ____
```

Chapter 3

CREATING A DIGITAL WATCH

Introduction

Now, having understood the FOR...NEXT commands, we can create a digital watch on the computer.

The Seconds

To begin with, we will concentrate on the display of the seconds.

? Look at the seconds on a digital watch. Do they start from Ø or 1?

? Is Ø displayed on the seconds after 1 minute ________(Yes or No)?

You can see that the seconds start from Ø and end at 59. (The watch alters the minutes instead of displaying 6Ø and the seconds are reset to Ø.)

Now we will begin to write the program in stages:

Firstly, in the centre of the screen, display the numbers from Ø to 59 in MODE Ø.

If the numbers are displayed to quickly, put a delay loop into the program.

Answer 16

☐ Now add a line to the program that will make the seconds return to Ø after reaching 59.

Answer 17

Setting the Correct Watch Rate

? Do the numbers on the screen appear faster or slower than those on a digital watch?

```
    FOR T=1 TO _____:NEXT T
```

☐ Get the program to work at the same rate as a digital watch i.e. change the delay loop so that the numbers printed on the screen appear exactly 1 second after each other.

Answer 18

The Minutes

As stated before, when the seconds on a digital watch reach 59, the minutes are incremented (Minutes=Minutes+1).

☐ Add lines to the program which display the minutes as well.

(Hint: you should add another FOR...NEXT loop.)

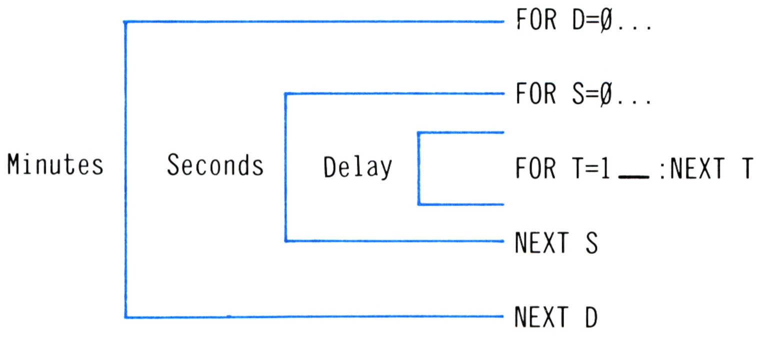
Note: before you run the program, make sure the screen looks like this:

```
         Ø : Ø
         ↑   ↑
     Minutes  Seconds
```

Answer 19

Nested Loops

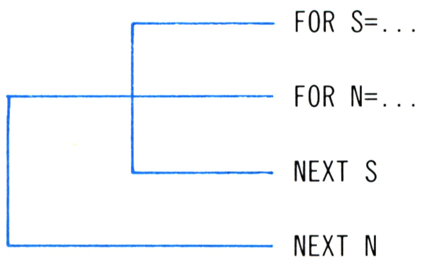
Let us stop for a moment and have a look at our program. You can see that there are 3 FOR...NEXT loops in the program:



Note: you can use FOR...NEXT loops in this way and they are said to be NESTED LOOPS.

Nested Loops

What about if we cross the FOR...NEXT loops. Is this of any use?



Answer 20

Task

If you would like to continue with the watch program, add in the hours and maybe even the date.

Shortening the NEXT Command

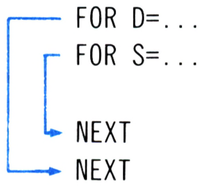
Up until now, we have followed the NEXT command with a variable e.g. NEXT S. However, the computer will accept the command without the variable, as shown in the following example:

```
    FOR T=1 TO 91Ø:NEXT
```

Is there any danger that the computer will get mixed up and not know which FOR to return to?

```
FOR D=...
FOR S=...
NEXT ?
NEXT ?
```

There isn't any danger because we have already said that loops cannot cross. Therefore, the computer will always perform the NEXT command on the last FOR command specified, even if you don't include the variable in the NEXT command.



```
FOR D=...
FOR S=...

NEXT
NEXT
```

The computer matches the last FOR in the program with the first NEXT.

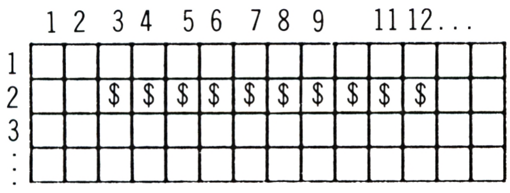
Chapter 4

PRINTING ON THE SCREEN

Printing Tasks

We will begin this chapter with some printing tasks:

☐ Write a program that will print the $ (dollar) character 1Ø times in MODE Ø, in the following way:



Note: only one $ should appear in the program.

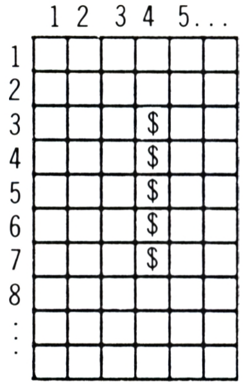
If you are having problems, complete the following program:

```
5  MODE Ø
1Ø FOR X=_____ TO _____
2Ø LOCATE _____,2:PRINT "$"
3Ø NEXT _____
```

Answer 21

If the dollars are printed to quickly, insert a delay loop and then run the program.

Now write a similar program that will print the dollars in the following positions:

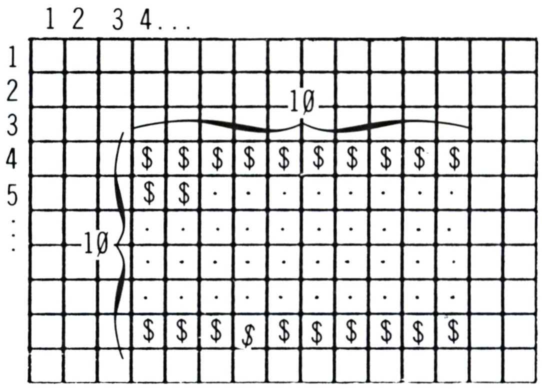


(Call the vertical variable Y.)

Answer 22

Task

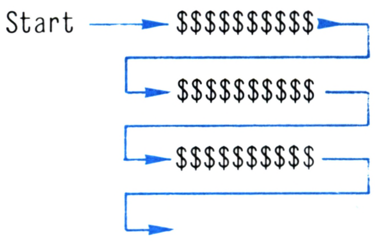
Write a program that produces the following pattern of dollars remembering that only one dollar is allowed in the actual program:



Hint 1

We want to write a program that displays the top line of dollars and then the second line of dollars etc until all ten lines are displayed.

The following diagram shows you how the program should work:



The diagram is showing you how the dollars are printed. When one line is printed, the computer goes to the next line and prints another line of dollars.

If you know how this is done, you can probably write the program, however, if you don't, look at the next hint:

Hint 2

There are two lines missing in the following program. Fill them in.

```
10 MODE 0
20 _____
30 FOR X=4 TO 13
40 LOCATE X,Y:PRINT "$"
50 NEXT X
60 _____
```

If you have filled them in correctly, run the program.

Answer 23

☐ Change the program so that the dollars measure 5 by 5 instead of 10 by 10.

☐ Now change the program so that the pattern is 15 by 15.

You should have made two changes to the program when altering the size of the pattern.

Task

Modify the program so that when you want to alter the size of the pattern, you need only change one line.

(Hint: you need to include another variable.)

Answer 24

Improving the Program

Instead of altering the program everytime you want to change the size of the pattern, let the computer ask you its size before printing the dollars.

To do this, we must learn a new command:

INPUT

☐ Add this line to the program currently in memory:

```
    12 INPUT L
```

and run the program.

A question mark (?) appears on the screen with the cursor next to it.

At this point, the computer is waiting for a number to put in L (INPUT L).

INPUT

☐ Type in a number (the pattern size) and press ENTER.

The program then continues and prints the dollars according to the size of L.

? How did this happen?

When the computer reached line 12, it stopped and displayed a question mark. It then waited for you to enter a value into variable L.

After entering the value, the computer put it into L and continued running the program. From there on, L equaled the value entered.

Task

Change the program so that after printing each pattern, the computer asks you for the size of the next pattern without erasing the previous one.

Only when you have entered the patterns size should the previous pattern be erased to make room for the new one. The program should continue like this.

Answer 25

Another Improvement

You can also choose the character which makes up the pattern (instead of having dollars all the time).

☐ Enter the following line:

```
14 INPUT A$
```

and change line 4Ø to:

```
4Ø LOCATE X,Y:PRINT A$
```

☐ Now run the program.

Again, the computer waits for you to input the patterns size, however, when you press ENTER, another question mark appears. The computer now wants to know the character which will create the pattern. (In the past, this character has been a $.)

Before we give you a detailed explanation, play around with the program by:

- Entering different characters.

and

- Entering graphics characters.

String Variables

Line 12 deals with the input of a numeric variable (L):

```
12 INPUT L
```

Line 14 deals with a variable of a different type, called a STRING VARIABLE:

String Variable

```
14 INPUT A$
```

? What is a String?

In fact, you have already used strings without knowing it. For example:

```
PRINT "2+5"
```

In this case, the computer doesn't add the numbers 2 and 5 but instead **prints** what is inside the quotation marks.

Note: everything inside the quotation marks is considered a string:

? In the following command:

```
PRINT "4*7=";4*7
```

which is the string and which is the expression?

Answer 26

? How many characters are there in the following strings:

```
"AMSTRAD"

"*25/7↑"
```

Answer 27

? What would happen if you told the computer to print the following string:

```
"BASIC"
```

and by mistake, told it to print:

```
BASIC
```

The computer would print Ø because it understood 'BASIC' as a Numeric Variable. (If you've forgotten what a Numeric Variable is, look at page 7.)

? What is a String Variable ?

If you understood numeric variables, you shouldn't have any difficulty understanding string variables:

☐ Type the following into the computer:

```
LET B$="XØR"
```

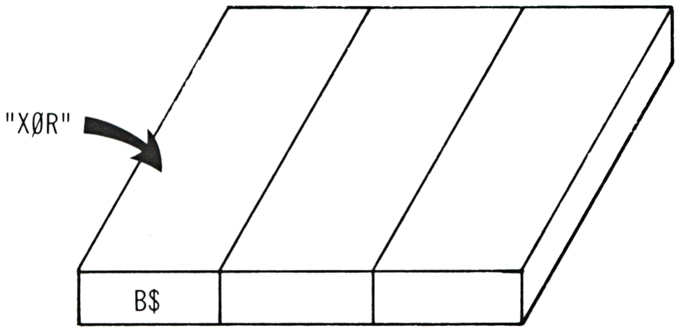
(Don't erase the program in memory.)

? What will happen if you instruct the computer to:

```
PRINT B$
```

☐ Try it.

When you LET B$="XØR", the computer puts the string "XØR" into a memory location and calls the memory location B$.



Now, when you PRINT B$, the computer looks for a memory location called B$ and prints its contents (XØR).

What will happen, in your opinion, if you type in:

```
LET B$=5
```

The computer informs you of a 'Type mismatch' error. That is to say, the variable (in this case B$ which is a string variable) and the value you what to store (the number 5) do not match.

Note: a string variable will only accept characters in quotation marks.

Will the computer accept LET B$="5"? Try it.

What happens when you try to put a string into a numeric variable, e.g. LET B="5".

What does the following program do:

```
500 X$="PAUL"
510 FOR N=1 TO 10
520 PRINT X$
530 NEXT
```

We have used these line numbers so as not to overwrite the program in memory.

Type the program in and then RUN 500. (This will execute the program starting from line 500.)

Deleting Program Lines

It is okay when you want to delete single lines from a program to simply type in the line number and press ENTER, but if you want to delete a lot of lines, it is easier to do it in one command. For example, we have finished with lines 500 to 530. Therefore we can instruct the computer to:

```
DELETE 500-530
```

DELETE

Now, when you display the program, you will see that ALL the lines between 5ØØ and 53Ø have been deleted from the program.

String Variable Names

The rules that apply to string variable names are the same as those for numeric variable names:

- It must begin with a letter.
- Either more letters or numbers may follow the first letter.

However, a string variable must end with a '$'. e.g. B$.

String Variable Names

Which of the following are illegal string variable names:

```
A$, 2A$, AAA$

B3, C$$, C*$
```

Answer 28

The Null String

Do you remember what happens when you print a numeric variable without previously defining it?

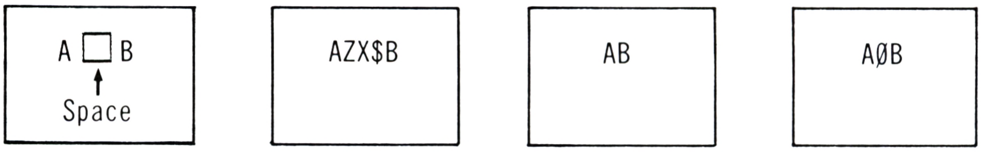
The computer prints Ø.

How does the computer react to an undefined string variable?

We will answer this question in stages:

Firstly, which of the following will appear when you instruct the computer to:

```
PRINT "A";ZX$;"B"
```



Try it.

Explanation

ZX$ is a string variable which hasn't had any value assigned to it.

When you try to print a string variable that has no value, the computer sets up a memory location (ZX$) but nothing is put in it.

The Null String

A string variable, such as ZX$, with nothing in it is known as a NULL STRING and is equivalent to saying ZX$="" with absolutely nothing between the quotation marks.

Back to the Program

Going back to the program, we wanted the computer to display a pattern of characters where the character (A$) and the pattern size (L) were input by the user.

When the computer reaches:

```
14 INPUT A$
```

it displays a question mark on the screen and waits for you to type in a character.

Note: you don't need to enter any quotation marks because the computer knows that it must be a string.

How does it know this?

The computer knows that your input is a string because in line 14, you specify a string when you end the variable name with a '$' (A$).

Each time the computer executes line 4Ø:

```
    4Ø LOCATE X,Y:PRINT A$
```

it prints the character that you input in a different position.

? What will happen if you put 2 characters into A$ e.g. A$="AB"

The Informative INPUT

The current program has two INPUT commands in it:

- One for the pattern size.
- One for the character.

You have probably noticed that the question marks that the computer displays are identical. This could be confusing to anyone who is not familiar with the program.

It is possible to print a message adjacent to the question mark which would state, quite clearly, what the computer is expecting. For example, change line 12 to this:

```
    12 INPUT "PATTERN SIZE";L
                    ↑        ↑
                 Message  Variable
```

Now run the program. The following should be displayed:

```
    PATTERN SIZE?■
```

and there is no mistake what the computer is waiting for.

❑ Stop the program running and enter a message for the second input. For example:

```
14 INPUT "CHARACTER";A$
```

Note: you may have noticed that it is possible to leave out the semi-colon in a PRINT instruction e.g.

```
PRINT "2+5" 2+5
```

is exactly the same as

```
PRINT "2+5";2+5
```

However, it is not possible to leave it out in the INPUT command.

❑ Try it out by changing line 14 and running the program.

RENUM

❑ Display the program currently in the computer's memory.

You can see that because we inserted lines between lines 1Ø and 2Ø, the program is a bit 'crowded' i.e. there is not much room left between lines 1Ø and 2Ø.

The Amstrad computer allows you to renumber program lines with the help of the RENUM command.

RENUM

❑ Type in:

```
RENUM
```

and after pressing ENTER, display the program.

The computer has renumbered all the program lines in steps of 1Ø. ie. first line is 1Ø, second line is 2Ø etc.

Look at the GOTO command on the last line of the program.

To which line is the program now directed? _________(12, 2Ø)?

Conclusion

When the computer executes the RENUM command, it also changes the line numbers specified in GOTO commands.

In our program, it changed line 12 to line 2Ø, and therefore, GOTO 12 became GOTO 2Ø.

Add lines to the program so that the computer prints the two INPUT messages on the top two lines of the screen and not underneath the pattern.

Answer 29

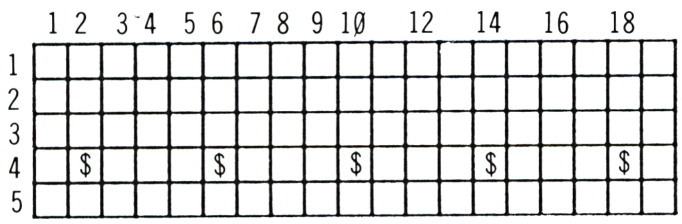
When you have finished, renumber the program again.

Chapter 5

ROBERTO THE ROBOT

Using the STEP Command

❑ Look at the diagram below:



❑ Complete the following program so that the computer prints the 5 dollars:

```
5 MODE Ø
1Ø X=_____
2Ø FOR N=1 TO _____
3Ø LOCATE _____,_____:_____
35 FOR T=1 TO 2ØØ:NEXT T
4Ø X=_____ + _____
5Ø NEXT N
```

Answer 30

Shortening the Program

It is possible to significantly shorten the program in memory by using the STEP command.

STEP

❑ Type in the following program remembering to erase the computer's memory first:

```
5 MODE Ø
2Ø FOR X=2 TO 18 STEP 4
3Ø LOCATE X,4:PRINT "$"
35 FOR T=1 TO 2ØØ:NEXT T
4Ø NEXT X
```

Now run the program. Is the same result produced?

What will happen if you change STEP 4 to STEP 2?

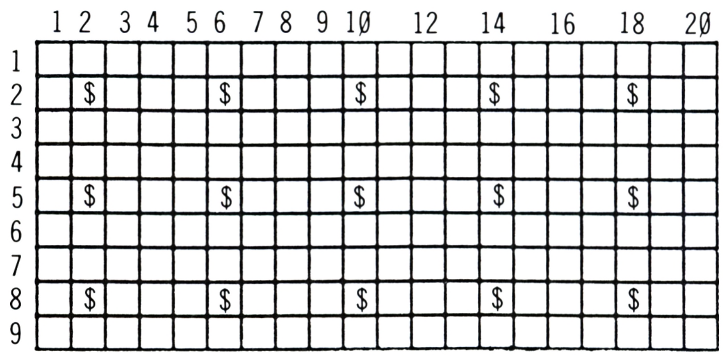
Conclusion

Including STEP in the FOR...NEXT command tells the computer to increase the loop by the amount specified. For example:

- STEP 4 tells the computer to increase the loop by ________ each time.
- STEP 2 tells the computer to increase the loop by ________ each time.

Task

Change the program so that the computer prints 3 lines of dollars, with two lines separating each of them.



Answer 31

Erase the program and continue.

What will the following program do:

```
1Ø FOR N=1 TO 4Ø STEP 5
2Ø PRINT N
3Ø NEXT N
```

After thinking about it, run the program.

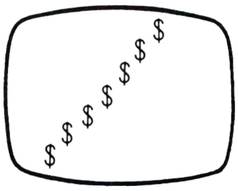
And what will happen when you run the following program:

```
1Ø FOR N=1ØØ TO Ø STEP -1Ø
2Ø PRINT N              ↑
3Ø NEXT N          Note the minus sign
```

This time, the computer printed the numbers in descending order i.e. from 1ØØ to Ø in steps of -1Ø.

Task

Use a negative STEP value to display the following drawing in MODE Ø:



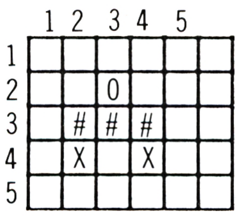
Answer 32

Change the program so that each '$' is a different colour.

Answer 33

Roberto the Robot

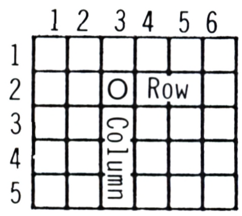
Please meet Roberto, younger brother of Robert from Unit 1.



Complete the following lines to display Roberto's head in row 2, column 3.

Note: the head is constructed from the letter 'O'.

```
5 MODE Ø
1Ø X=_____
2Ø Y=_____
3Ø LOCATE X,Y:_____
```



Now add the body (###).

The body starts one line _________(above, below) the head and one column to the _________(left, right) of it.

Therefore, the command to display the body will be:

```
4Ø LOCATE X-_____,Y+_____:
PRINT _____
```

Note the minus sign

Complete the line and run the program.

If you find it difficult, look at Answer 34. **Answer 34**

Do the X and Y values change after the computer executes line 4Ø?

Now add the line to display the legs. **Answer 35**

Task

Change the program so that the computer asks you in which column (X) and in which row (Y) you want Roberto to appear. The position input will be Roberto's head position and the rest of Roberto will be drawn from this point.

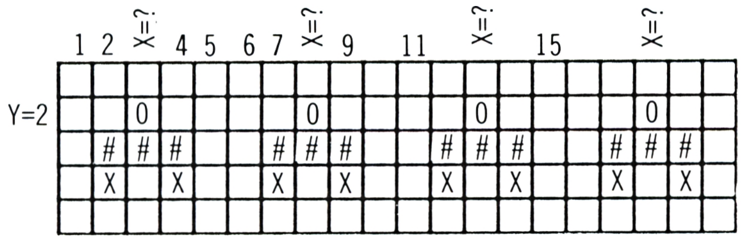
When the robot has been printed, the computer asks you for another position.

Answer 36

Reproducing Roberto

Up until now, we have displayed Roberto only after entering the values of X and Y.

Develop a program which will display Roberto four times along the width of the screen in the following positions:



(Hint: use FOR...NEXT...STEP.)

Answer 37

Change the program so that three rows of 4 robots are displayed.

Hint: look at page 50.

Answer 38

Improvements

- Display Roberto diagonally.
- Display Roberto across both diagonals i.e. he meets in the middle of the screen.

CHR$

Change the line that prints Roberto's head to:

```
   30 LOCATE X,Y:PRINT CHR$(224)
```

and run the program.

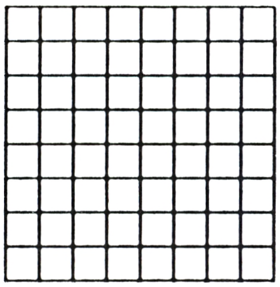
If you did that correctly, you'll see that the computer has printed the robot with a smiling face.

How did this happen?

For each character that the computer has, whether it be a letter, number or graphics character, there is a character number which is permanently stored in the computer's memory.

Open your Amstrad manual to the character set page.

You should be looking at a lot of enlarged characters where each character is displayed in an 8 by 8 grid, like this:



Find character 224, the smiling face.

What will happen if instead of using character 224, you use character 225. Try it.

Is Roberto sad now?

And what will happen if you use character 79?

Explanation

CHR$(79) is equal to the letter 'O'.

CHR$

Note: a lower case 'o' has a different character value to an uppercase 'O'.

Restore the smiling face back to the robot.

Now, change the robot's body so that instead of having the 'hash' character (#), we have character 2Ø7 which is a graphics character.

Note

Since CHR$(2Ø7) is a graphics character and cannot be printed in quotes, it must be printed 3 times. **Answer 39**

Changing his Legs

☐ Choose two suitable graphics characters from the list of characters in the manual, to replace the legs ('X') in the robot. **Answer 40**

Colouring Roberto

☐ Add lines to the program so that each robot appears in a different colour.

Task

Look at the graphics characters 248, 249, 25Ø and 251.

You can see that the same shape, a small man, is displayed in all four characters, however, in each, it is displayed differently.

Write a program that will make the small man 'dance' in the centre of MODE 1 i.e. you must continuously print the characters between CHR$(248) and CHR$(251). **Answer 41**

Challenge

Write a program that will display, in MODE 1, all the characters from 33 to 255.

CHR$(13)

On page 20, we used CHR$(13) to define the number pad. We can now explain why.

Although CHR$(13) in not a visible character, it is what we call a 'Control Character'.

The Amstrad has 32 control characters, (CHR$(Ø) through to CHR$(31)) and each one performs a different function. Therefore, when you print a control character, rather than a shape appearing on the screen, the computer performs a function.

Therefore, if you type in the following command:

```
    KEY 137,"LIST"+CHR$(13)
```

it is equivalent to putting ENTER at the end of the LIST command because CHR$(13) represents the ENTER function.

Note: a list of Control Characters can be found in the manual just before the visible characters.

Chapter 6

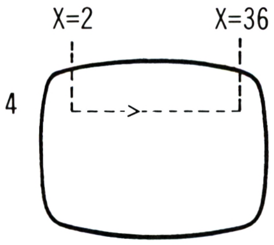
ANIMATION

Do you remember Robert doing his exercises in Unit 1?

In this Chapter, we will use the commands we have learnt, to move shapes on the screen.

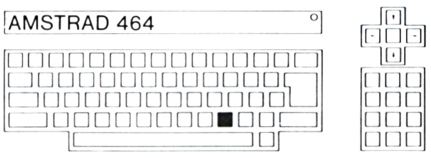
Task 1

Write a program that will move the greater than sign ('>') across the screen (MODE 1) from column 2 to column 36 in row 4.



When the '>' sign reaches column 36, it repeats the process again and again.

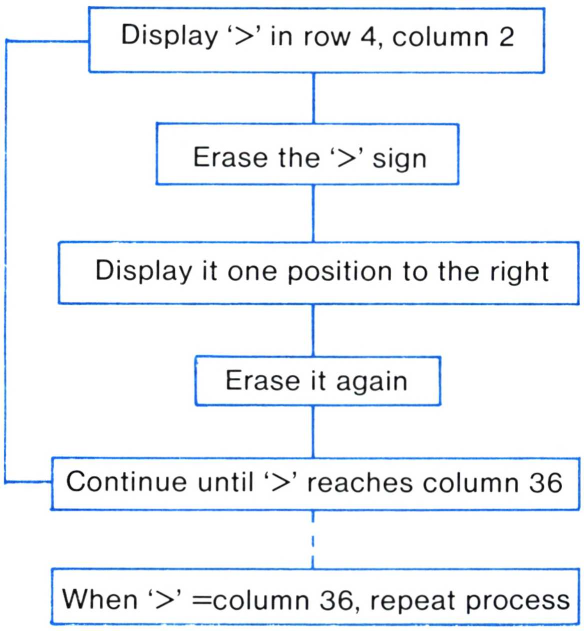
Note: the '>' sign is found here on the keyboard:



You should know all the necessary commands to write this program, however, if you have any difficulty, look at the following hint.

Hint

The diagram below shows how the program should be structured (written):

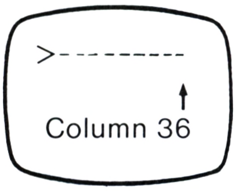

Display '>' in row 4, column 2
Erase the '>' sign
Display it one position to the right
Erase it again
Continue until '>' reaches column 36
When '>' =column 36, repeat process

You should now be able to write the program, however, if you can't, look at Answer 42.

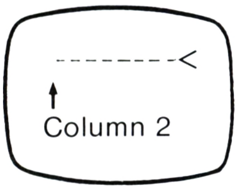
Answer 42

Task 2

Change the program so that when the '>' sign reaches column 36:



the shape changes to a less than ('<') sign and the less than sign then makes its way back to column 2.



- At the points where the '>' and '<' signs change, display graphics character 143. i.e. the signs look like there bouncing off the two blocks.

- You will find the less than sign next to the greater than sign on the keyboard.

When writing this program, make sure that:

- the signs don't erase the blocks,

- the signs continuously move back and forth,

- the walls don't flicker,

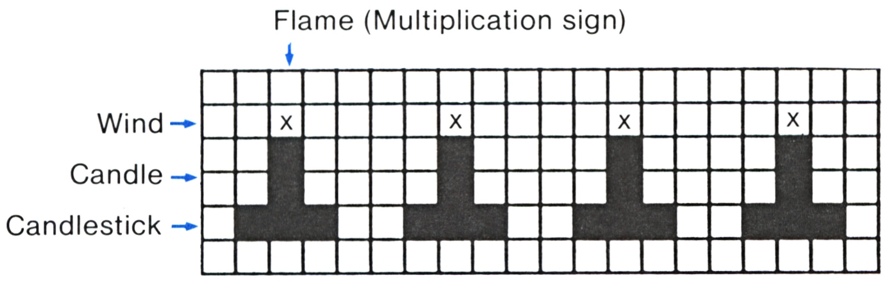
- the signs bounce off the blocks i.e. the signs finish one position before the block.

If you wish, you can colour the blocks and the signs to make the program more interesting.

Answer 43

Task 3

Write a program that displays 4 candles. When the candles are lit, a wind appears ('>') from the left and blows the candles out.



Below are some hints on how to write the program:

- the wind moves in the same way as the signs in the previous program,
- the flames are constructed from the multiplication sign,
- the program reproduces the candles in the same way as Roberto was reproduced in the previous chapter.

If you wish, you can include colour.

Answer 44

Challenge

Modify the program so that each time the wind blows the candles out, they re-light themselves and the wind tries to blow them out again.

Chapter 7

MORE PRINTING TECHNIQUES

In this chapter, we will show you some more printing techniques which will improve the 'display' of your programs.

Write a program to do the following:

- The computer asks you (INPUT) for a name that you would like to display in a box.

- When you have input the name, the computer displays it in the centre of the screen and draws a box around it. The size of the box is dependent on the length of the name.

For example:

```
********
*      *
* PAUL *
*      *
********
```

```
***********
*         *
* MICHAEL *
*         *
***********
```

LEN

In order to write this program, we must learn a new command; the LEN command.

LEN

LEN is short for LENgth.

To understand how the LEN command works, instruct the computer to:

```
    PRINT LEN("ABCDE")
```

On pressing ENTER, the computer displays the number ________.

This number represents the number of characters in the string.

What will the computer display on the screen after executing the following program:

```
    2Ø X1$="A1B2C3"
    3Ø PRINT LEN(X1$)
```

In this case, the LEN command displays the number of characters in X1$.

Note: the LEN function will only work on strings and not numeric variables.

Now try to write the box program using the LEN command. If you can't do it on your own, follow the stages below:

Stage 1

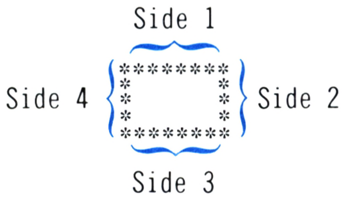
First of all, you should know how to draw boxes to defined sizes i.e. the length and width are input and the box is drawn from them.

```
********
*      *
*      *
*      *
********
```

The size of the box above is 8 by 5 (8 asterisks in length by 5 asterisks in width).

Write a program to draw this box in MODE Ø, where the top left asterisk is in row 3, column 3.

How do you do this?



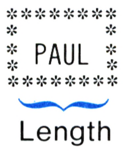
There are a number of ways. You may like to draw side 1 first and then sides 2, 3 and 4, or you could draw sides 1 and 3 simultaneously and then sides 2 and 4 simultaneously. **Answer 45**

Add lines to the program so that the computer asks you for the length of the box. **Answer 46**

Stage 2

Now we will go on to displaying the name.

The main problem is displaying a box whose size is dependent on the length of the name.



```
********
*      *
* PAUL *
********
```

The name is made up of 4 characters.

The box is ________(2, 4, 8) asterisks in length.

Therefore:
Box Length = Length of Name+________(2, 4, 8). **Answer 47**

Does the 'READY' message ruin the display?

Modify the program so that when you have typed in the name to be displayed, the screen is cleared leaving just the boxed name on the screen. This is then held continuously on the screen. **Answer 48**

Improvements

- The computer asks you from which character you want the box constructed.
- The computer asks you what background and what character colour you require.

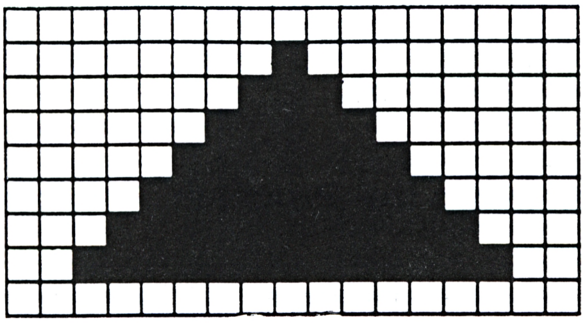
Chapter 8

CONSTRUCTING SHAPES

This Chapter is designed to help you produce shapes on the screen. It is quite a difficult chapter so don't worry if you can't follow it all.

Task 1: Constructing a 'Pyramid'

Write a program to display the following pyramid:



The program should be written with only one LOCATE command in it.

If you feel confident to tackle this task, do so, however, we expect most of you to follow the stages below.

Stage 1

? What does the following program do?

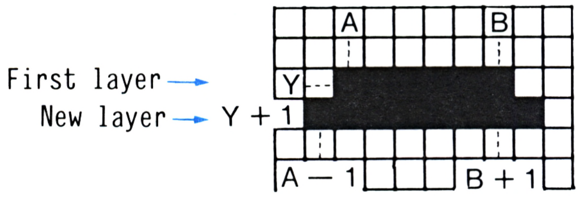
```
10 MODE 1
20 A=15:B=25:Y=5
30 FOR X=A TO B
40 LOCATE X,Y:PRINT CHR$(143)
50 NEXT X
```

After thinking about it, type the program in and run it.

Can you write the pyramid program now?

Stage 2

Change the program in memory so that another layer is added below the first layer, like this:



As you saw on page 65, the pyramid is constructed from increasing size layers. Therefore, if you improve this process, you will have solved the problem.

Answer 49

Improving the Program

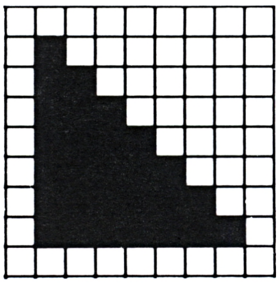
Change the program so that the computer asks you which character the pyramid is to be constructed from.

Answer 50

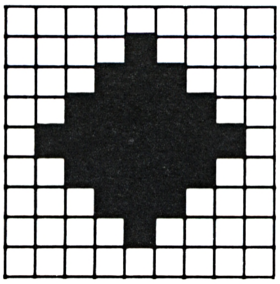
If you wish, add some colour.

What must you do to the program in memory to produce a shape like this?

Answer 51



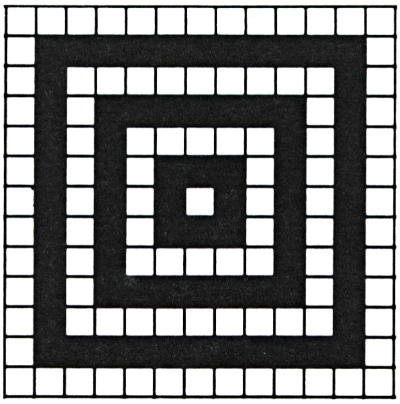
What about the following shape:



We'll let you do this on your own.

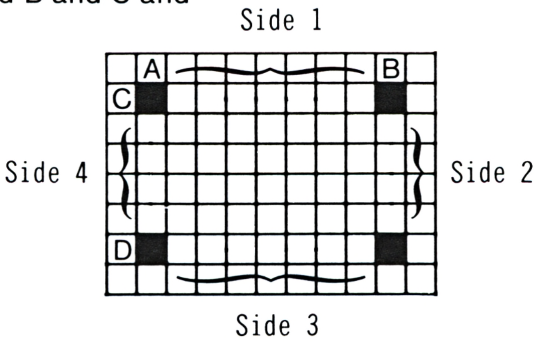
Task 2: Concentric Squares

Write a program to display the following concentric squares.



The program should display the inner square first and then work its way out. If your not sure what to do, continue reading.

The first thing we want to do is to construct a square between the points A and B and C and D.



Stage 1

Write a program to draw side 1, remembering to give values to A, B and C.

Answer 52

Stage 2

☐ Add lines to the program to display side 2. (Give a value to D as well.) **Answer 53**

Stage 3

☐ Now draw side 3.

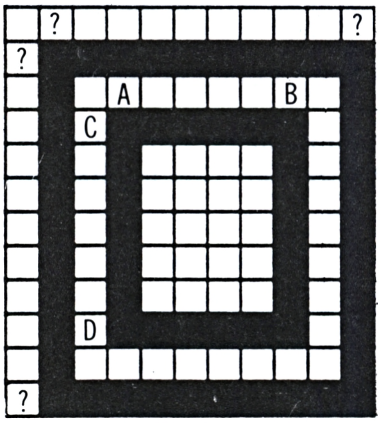
Note: you must use a negative STEP. **Answer 54**

Stage 4

Finish off the last side on your own and run the program. If you cannot do it, go through the stages again making sure you understand everything that we are doing.

Stage 5

What must happen to A, B, C and D for the program to draw three larger squares?



☐ Enter the appropriate lines. **Answer 55**

Improving the Program

- Add colour to the program.

- Modify the program so that when one square has been displayed, it is rubbed out and another 'larger' square is displayed.

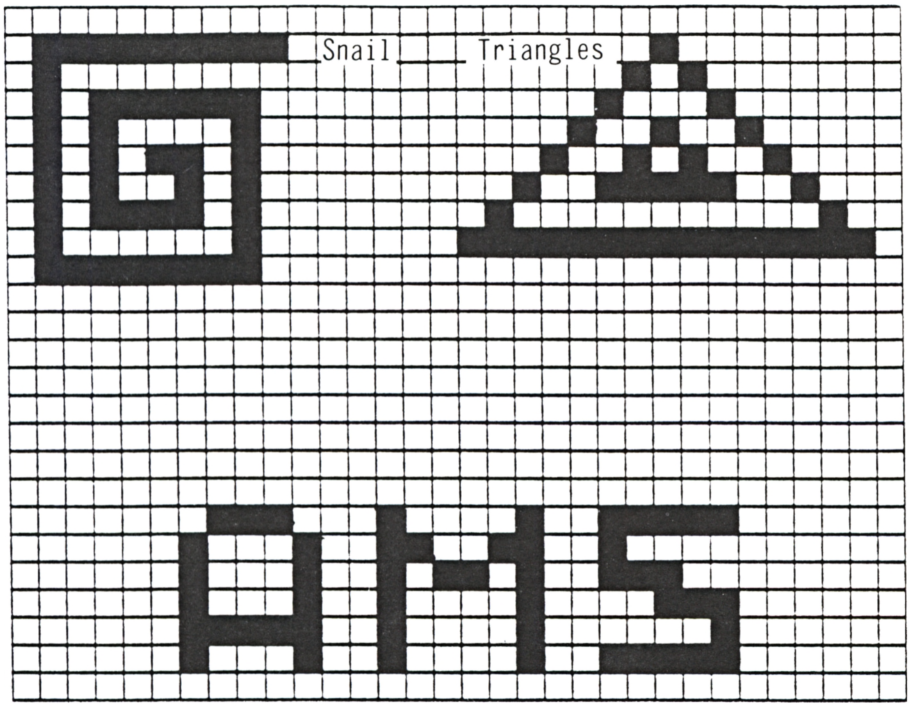
- Change the size and position of the squares.

- The computer constructs the squares from the character of your choice.

Challenge

Below are three drawings. Try to write the programs that display each of the shapes.

Note: this may take you a long time but it will be well worth your while in the end.



SUMMARY INDEX

Below is a list of the commands and techniques taught during this unit.

Go over each one of them and check if you remember them.
(Explanations are given on the pages stated in the brackets.)

Conclusion

You have now come to the end of Amstrad Unit 2 and should have acquired some good programming techniques.

The next booklet in this series, Unit 3, is called 'Computer Games' and shows you how to develop computer games. At this stage, the games are quite simple, however, the knowledge obtained from Unit 3 will stand you in good stead for the more complex games that come up in later units.

ANSWERS

Answer 1

The computer informs you that you have divided by 0 because the variable F was not previously defined and so received the value of Ø. Therefore, the computer printed:

1.7Ø141E+38 = 1.70141*1Ø^{38}

This is the largest number that the computer can deal with.

Note the way that the computer displays the 'to the power of'. For example, 7E+12 equals 7*1Ø^{12} or 7 with 12 zeros.

Answer 2

The illegal variable names are:

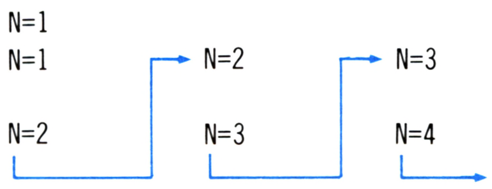
```
2D, 25, PRINT
```

Answer 3

```
1Ø MODE Ø
2Ø N=1
3Ø LOCATE 1Ø,1Ø:PRINT N
35 FOR T=1 TO 3ØØ:NEXT T
4Ø N=N+1
5Ø GOTO 3Ø
```

Answer 4

```
1Ø MODE Ø
2Ø N=1
3Ø LOCATE 1Ø,1Ø:PRINT N
35 FOR T=1 TO 3ØØ:NEXT T
4Ø N=N+1
5Ø GOTO 3Ø
```



Answer 5

Change this line:

```
4Ø N=N+2
```

Answer 6

Change this line:

```
1Ø FOR N=1 TO 8
```

Answer 7

The computer will execute them twenty times.

Answer 8

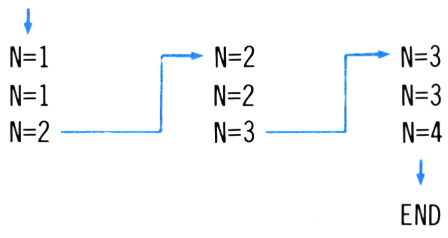
```
1Ø MODE 1
2Ø FOR B=1 TO 7
3Ø LOCATE B,B:PRINT "A"
4Ø NEXT B
```

Answer 9

```
1Ø MODE 1
2Ø FOR N=1 TO 5
3Ø LOCATE N,N:PRINT "PAUL"
4Ø NEXT N
```

Answer 10

```
1Ø FOR N=1 TO 3
2Ø PRINT N
3Ø NEXT N
4Ø PRINT "END"
```



When N is greater than 3, the computer exits from the FOR...NEXT loop and continues to line 4Ø.

Answer 11

Change this line:

```
20 FOR T=1 TO 2000
```

Answer 12

The answer is in the following 2 lines:

```
20 FOR T=1 TO X*X:NEXT T
30 X=X+1
```

It is called the ‘tired program’ because X gets larger and larger. Therefore, each time the FOR...NEXT loop is executed,it takes longer and longer to complete i.e. on the first loop, X equals 1 and produces a delay from 1 to 1; on the second loop, X equals 2 and produces a delay from 1 to 4 etc.

Answer 13

```
10 MODE 0
20 FOR N=0 TO 15
30 INK 0,N
40 FOR T=1 TO 1000:NEXT T
50 NEXT N
60 GOTO 20
```

Answer 14

Add this line:

```
35 BORDER 15-N
```

Answer 15

Change this line:

```
40 SOUND 1,5*N,100
```

Answer 16

```
10 MODE 0
20 FOR S=0 TO 59
30 LOCATE 10,10:PRINT S
40 FOR T=1 TO 500:NEXT T
50 NEXT S
```

Answer 17

Add this line:

```
60 GOTO 20
```

Answer 18

Change this line:

```
40 FOR T=1 TO 910:NEXT T
```

Answer 19

Add these lines:

```
15 FOR D=0 TO 59
30 LOCATE 7,10:PRINT D;":";S
55 NEXT D
```

Answer 20

You cannot cross loops in this way and if you do, an error will occur. Try it.

Answer 21

```
5  MODE 0
10 FOR X=3 TO 12
20 LOCATE X,2:PRINT "$"
30 NEXT X
```

Answer 22

```
5  MODE 0
10 FOR Y=3 TO 7
20 LOCATE 4,Y:PRINT "$"
30 NEXT Y
```

Answer 23

```
10 MODE 0
20 FOR Y=4 TO 13
30 FOR X=4 TO 13
40 LOCATE X,Y:PRINT "$"
50 NEXT X
60 NEXT Y
```

Answer 24

Add these lines:

```
12 L=10
20 FOR Y=4 TO 3+L
30 FOR X=4 TO 3+L
```

Answer 25

Add these lines:

```
18 MODE 0
70 GOTO 12
```

Answer 26

$$\texttt{PRINT }\underset{\substack{\uparrow\\ \text{String}}}{\texttt{"4*7="}}\texttt{;}\underbrace{\texttt{4*7}}_{\substack{\uparrow\\ \text{Expression}}}$$

Answer 27

There are 7 characters in "AMSTRAD" and 6 characters in "*25/7↑".

Answer 28

The following are illegal String Variable names:

```
2A$, B3, C$$, C*$
```

Answer 29

Add these lines:

```
15  LOCATE 1,1
1ØØ GOTO 15
```

Answer 30

```
1Ø X=2
2Ø FOR N=1 TO 5
3Ø LOCATE X,4:PRINT "$"
4Ø X=X+4
```

Answer 31

```
1Ø FOR Y=2 TO 8 STEP 3
3Ø LOCATE X,Y:PRINT "$"
5Ø NEXT Y
```

Answer 32

```
5  Y=5
1Ø MODE Ø
2Ø FOR X=15 TO 5 STEP -1
3Ø LOCATE X,Y:PRINT "$"
4Ø Y=Y+1
5Ø NEXT X
```

Answer 33

```
25 PEN X-5
```

Answer 34

```
4Ø LOCATE X-1,Y+1:PRINT "###"
```

Answer 35

```
5Ø LOCATE X-1,Y+2:PRINT "X X"
```

Answer 36

Add these lines:

```
7  LOCATE 1,1
1Ø INPUT "X=";X
2Ø INPUT "Y=";Y
6Ø GOTO 7
```

Answer 37

```
5  MODE Ø
7  LOCATE 1,1
1Ø FOR X=3 TO 18 STEP 5
2Ø Y=2
3Ø LOCATE X,Y:PRINT "O"
4Ø LOCATE X-1,Y+1:PRINT "###"
5Ø LOCATE X-1,Y+2:PRINT "X□X"
6Ø NEXT X
```

Answer 38

Add these lines:

```
2Ø FOR Y=2 TO 1Ø STEP 4
55 NEXT Y
```

Answer 39

Change this line:

```
40 LOCATE X-1,Y+1:PRINT CHR$(2Ø7);CHR$(2
Ø7);CHR$(2Ø7)
```

Answer 40

Change this line:

```
5Ø LOCATE X-1,Y+2:PRINT CHR$(211);"□";CH
R$(2Ø9)
```

Answer 41

```
1Ø MODE 1
2Ø FOR N=248 TO 251
3Ø LOCATE 2Ø,1Ø:PRINT CHR$(N)
4Ø FOR T=1 TO 5ØØ:NEXT T
5Ø NEXT N
6Ø GOTO 2Ø
```

Answer 42

```
1Ø MODE 1
2Ø FOR X=2 TO 36
3Ø LOCATE X,4:PRINT ">"
4Ø FOR T=1 TO 4Ø:NEXT T
5Ø LOCATE X,4:PRINT "□"
6Ø NEXT X
7Ø GOTO 2Ø
```

Answer 43

Add these lines:

```
15 LOCATE 1,4:PRINT CHR$(143)
17 LOCATE 37,4:PRINT CHR$(143)
7Ø FOR X=36 TO 2 STEP -1
8Ø LOCATE X,4:PRINT "<"
9Ø FOR T=1 TO 4Ø:NEXT T
1ØØ LOCATE X,4:PRINT "□"
11Ø NEXT X
12Ø GOTO 2Ø
```

Answer 44

```
1Ø MODE 1
2Ø FOR X=2Ø TO 35 STEP 5
3Ø LOCATE X,7:PRINT "*"
4Ø LOCATE X,8:PRINT CHR$(143)
5Ø LOCATE X,9:PRINT CHR$(143)
6Ø LOCATE X-1,1Ø:PRINT CHR$(143);CHR$(14
3);CHR$(143)
7Ø NEXT X
8Ø FOR X=1 TO 39
9Ø LOCATE X,7:PRINT ">"
1ØØ FOR T=1 TO 4Ø:NEXT T
11Ø LOCATE X,7:PRINT "□"
12Ø NEXT X
```

Answer 45

```
1Ø MODE Ø
2Ø FOR X=3 TO 1Ø
3Ø LOCATE X,3:PRINT "*" }  Sides 1 and 3
4Ø LOCATE X,7:PRINT"*"  }
5Ø NEXT X
6Ø FOR Y=4 TO 6
7Ø LOCATE 3,Y:PRINT "*"  }  Sides 2 and 4
8Ø LOCATE 1Ø,Y:PRINT "*" }
9Ø NEXT Y
```

Answer 46

Add these lines:

```
15 INPUT "L=";L
2Ø FOR X=3 TO 2+L
8Ø LOCATE 2+L,Y:PRINT "*"
```

Answer 47

Add these lines:

```
15 INPUT "NAME=";A$
17 LOCATE 5,5:PRINT A$
2Ø FOR X=3 TO 2+LEN(A$)+4
8Ø LOCATE 2+LEN(A$)+4,Y:PRINT "*"
```

Answer 48

Add these lines:

```
16  CLS
1ØØ GOTO 1ØØ
```

Answer 49

```
1Ø MODE 1
2Ø A=2Ø:B=2Ø
3Ø FOR Y=1 TO 2Ø
4Ø FOR X=A TO B
5Ø LOCATE X,Y:PRINT CHR$(143)
6Ø NEXT X
7Ø A=A-1:B=B+1
8Ø NEXT Y
```

Answer 5Ø

Add these lines:

```
5 INPUT A$
5Ø LOCATE X,Y:PRINT A$
1ØØ GOTO 5
```

Answer 51

Change this line:

```
70 B=B+1
```

Answer 52

```
10 MODE 1
20 A=19:B=21:C=9
30 FOR X=A TO B
40 LOCATE X,C:PRINT CHR$(143)
50 NEXT X
```

Answer 53

Add these lines:

```
20 A=19:B=21:C=9:D=11
60 FOR Y=C TO D
70 LOCATE X,Y:PRINT CHR$(143)
80 NEXT Y
```

Answer 54

Add these lines:

```
90  FOR X=B TO A STEP -1
100 LOCATE X,D:PRINT CHR$(143)
110 NEXT X
```

Answer 55

Add these lines:

```
25  FOR N=1 TO 4
120 FOR Y=D TO C STEP -1
130 LOCATE A,Y:PRINT CHR$(143)
140 NEXT Y
150 A=A-2:B=B+2:C=C-2:D=D+2
160 NEXT N
```

Notes

This series of self-instruction books will teach you the secrets of writing programs in BASIC on your AMSTRAD computer.

Unit 1: **FIRST STEPS IN BASIC**
Starting with the first things every programmer need to know, you will learn to issue commands to the computer, as well as writing and running programs. By the end of the unit you'll be able to make your computer perform useful and interesting tasks.

Unit 2: **EXPLORING BASIC**
This unit teaches you the most important concepts of BASIC: numeric variables, string variables, FOR . . . NEXT and IF . . . THEN statements, and much more. You'll create a digital computer clock, and interesting graphics programs including animation.

Unit 3: **COMPUTER GAMES**
In this unit you will learn to develop various computer video games. As you progress through the unit, new programming concepts such as random numbers will be introduced. By the end you'll have considerable programming skills.

And after that . . . we're planning more units to deal with special subjects, such as 3D graphics, machine code and more.

£3.95

ISBN 1-85181-002-1

9 781851 810024

GLENTOP
PUBLISHERS LIMITED

Glentop Publishers Ltd.,
Standfast House, Bath Place,
High Street, Barnet, Herts. EN5 1ED
Tel: 01-441 4130

AMSTRAD 464, 664 & 6128

EXPLORING BASIC

WATSON'S NOTES VOL 2