

GUARANTEE

Your ZX PRINTER is covered by a 12 month comprehensive guarantee, valid in UK only.

RETURNS/INFORMATION

Should you need to return your ZX PRINTER for any reason, please send it to the following address together with an explanatory letter.

SINCLAIR RESEARCH (JBS)
STANHOPE ROAD
CAMBERLEY
SURREY
GU15 3PS

For any other information please write to the above address or telephone (0276) 21282.

© Sinclair Research 1981

Cedric 30-4-82

Printed by The Langens Press Ltd, Union and London



ZX Printer Operating Instructions

INTRODUCTION

The Sinclair ZX printer gives a permanent record of anything the computer can print on the TV screen. In addition, graphics always "join up" if done in several goes, so that long programs, long tables of results and elaborate patterns and graphs can be printed out without a break.

Supplied with the printer are these instructions and 1 reel of paper. You will also need either a ZX81, or a ZX80 with the replacement ROM: the standard version of the ZX80 cannot be used. Chapter 20 of the ZX81 instruction book gives the BASIC commands necessary to use the printer. You will also need a 1.2A power supply which will have been supplied either with your computer or your printer.

CONNECTING UP

Before connecting anything to your computer, pull its power plug out. If you have a RAM pack, unplug that too. The edge socket on the end of the printer's cable now pushes onto the back of the computer through the rectangular slot – which is on the left of the ZX80 and on the right of the ZX81 (looking from the front). The printer will work perfectly well without a RAM pack, but if you have one an extra plug has been provided behind the printer's socket so that the two may be stacked together. Just push the RAM into the back of the socket housing the same way as if you were connecting it to the computer. The printer should now sit neatly on the right hand side of the ZX80/81.

IMPORTANT: Since the printer takes a lot of power, a new power supply is provided which *must* be used in place of the old computer supply, and which plugs into the same socket on the ZX80 or ZX81. No harm will be done using the replacement supply all the time, even if the printer isn't connected.

Note – since you shouldn't connect the printer without switching the computer off first, it's a good idea to have the printer always plugged in just in case – otherwise you'll have to save your program on tape while you fiddle about with plugs and sockets.

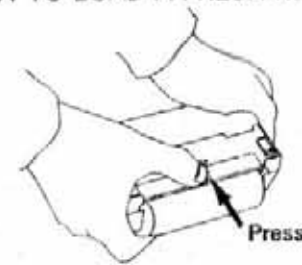
See LPRINT inside back cover

THE PAPER

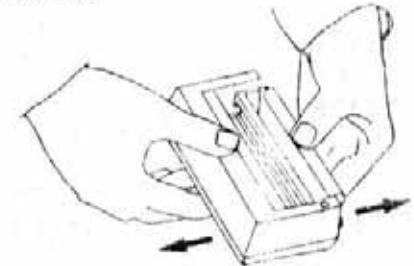
The ZX printer uses its own special metallized paper, which can be supplied by Sinclair Research. Other metal-coated papers will work, but may give

smeared, tuzzy or faint printing – others contain cadmium in the coating and could be a health risk. Non-metallized paper won't work at all.

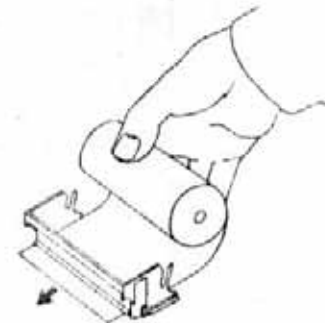
HOW TO LOAD A FRESH REEL OF PAPER.



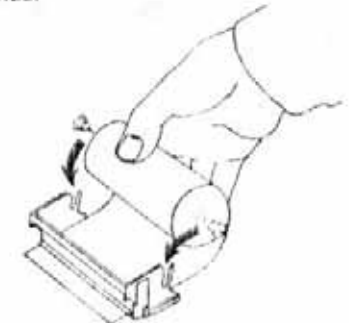
Press on the underneath of the reel carrier to disengage it from the printer.



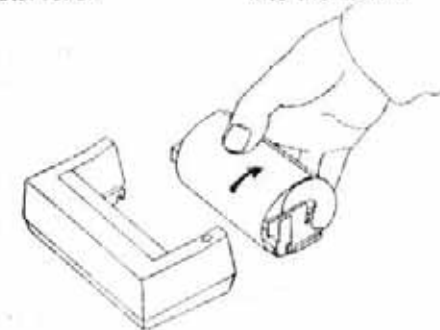
With the reel carrier pulled out of the printer, take out the empty cardboard tube – but save the plastic hubs in the ends.



Thread the end of the new reel underneath the metal roller.



Push the hubs into the reel, and clip it into the carrier.



Pull some of the paper through, hold the carrier as shown and clip it back into the printer body.

This may be done while the printer and computer are switched on, so long as care is taken not to disturb the connections.

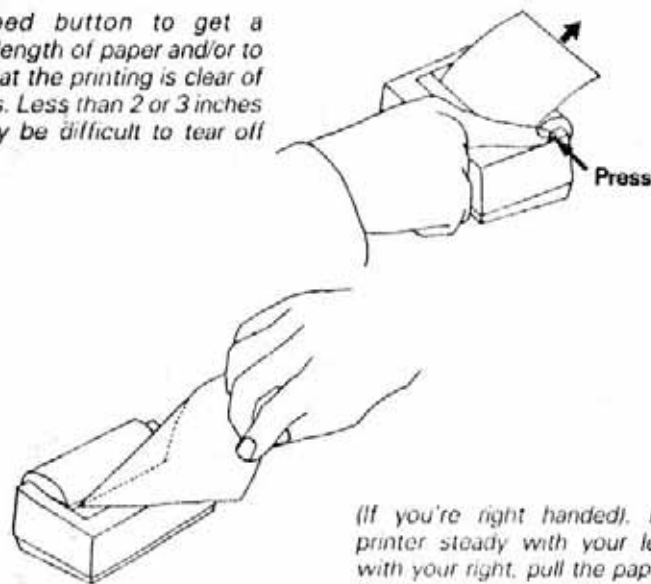
THE FEED BUTTON

This is the printer's only control, situated to the right of the paper reel. This feeds paper out without printing anything to give a blank between printouts or when tearing off. The feed button doesn't communicate with the computer, so it can be pressed at any time, even when the ZX80 or ZX81 is in the middle of a program. If you press the button whilst the printer is actually printing, however, the computer will be overridden for as long as the button is held down and you'll get a blank – no harm will be done, but the printout during that time will be lost.

TEARING OFF

Sections of the paper may be torn off using the printer's serrated edge – next to where the paper comes out.

Use the feed button to get a manageable length of paper and/or to make sure that the printing is clear of the serrations. Less than 2 or 3 inches of paper may be difficult to tear off properly.



(If you're right handed). Hold the printer steady with your left hand; with your right, pull the paper across the serrations at a fairly steep angle. Don't try to pull the paper squarely across the edge all at once.

Practice makes perfect – if you're in any doubt about the above procedure try tearing off a few strips of blank paper first.

CLEANING

After you've been through a whole reel, you'll find that some of the black material has been scraped off the paper. Whilst you have the reel carrier out, the printer should be cleaned. The black deposit should be removed from the recess where the reel carrier goes especially around the slot just underneath the serrations, and from the rubber drive roller. Use only tissues or a soft brush – chemicals are definitely out. Some of them will attack the plastic and may reduce your printer to a shapeless black blob.

Also, don't poke anything into any of the apertures, as the electronics or mechanics may be damaged, and don't try to take the printer apart. Dismantling is easy, reassembly may be more difficult.

ZX PRINTER TECHNICAL APPENDIX

This section is written for people who wish to use their printer with machine code programs to do things like high resolution graph plotting, alternative character sets, etc. and it assumes a knowledge of Z80 hardware and software. For printing out ZX81 characters from a BASIC program you don't need to understand this section at all.

GENERAL PRINCIPLE OF OPERATION

The printer functions in rather the same way as a TV picture, i.e. by scanning from left to right. A conductive stylus is pulled across the paper at high speed, and where a black dot is wanted a pulse of current is passed through the stylus. This evaporates the aluminium coating on the paper, and allows the black backing to show through. To avoid the need to return the stylus quickly to the left hand edge of the paper, there are in fact two styli, mounted on a moving belt, which follow each other in quick succession. The belt and the paper feed roller are both driven continuously whilst printing; so that when the next stylus comes round the paper has been moved up ready for the next line.

In order that the styli always stop off the paper, the BASIC commands slow the motor down for the last two scans before stopping – this makes sure that graphics always "join up" if done in several goes, and this feature should be put in any machine code program that requires the printer to stop whilst the computer thinks about the next line. At full speed, each pen is on the paper for about 32ms, and off it for about 16ms.

Since the speed of printing can vary, an encoder disc is used to give 256 pulses across a 92mm printing width (allowing 4mm margins). These pulses

are synchronised with the stylus hitting the paper so that the printing is always vertical.

There is about 60µs between the starts of successive pulses at full speed.

THE INTERFACE

The printer is wired as a Z80 I/O port, selected by A2 being at a low level – no other address lines are recognised. So to send information to the printer, use the Z80 command

OUT (FB),A – opcode D3 FB

assuming the data is in register A. The data bits have the following meanings:

- (D2) High level means stop the motor, low means start it.
- (D1) High level means put the motor on slow speed – this line is overridden by D2 being high.
- (D7) High level applies power to the stylus.

All these lines remain in the state they were at last, until new data is sent to the printer. At switch on, or after pressing the feed button, D1 and D7 are set low; D2 is left high once feed is finished. The other data lines aren't used.

To fetch information from the printer, the Z80 instruction: IN A, (FB) – opcode DB FB; will put the data into the accumulator. These bits are used:

- (D6) Will be read as low if the printer is there, high if it isn't, and is used solely to check the printer is connected
- (D0) This line is the signal from the encoder disc.
- (D7) This line is high when the stylus hits the paper.

D0 and D7 are both latched so that they remain high until the computer writes something to the printer. So even if you don't make use of the information you've read in, you should use an output instruction (with appropriate data) to reset the latches until the next signal. These bits may be in either state on switch on, and aren't affected by the feed button. The paper detect signal is also used internally by the printer to make sure that the styli stop off the paper. Note that if power is applied to the stylus, the paper signal will go high even if the printer is between scans, and so the stylus must be turned off before attempting to detect the edge of the paper.

SOME PROGRAMS

Here are a few example programs for experimenting with. "Text Edit" (No. 1) uses the printer rather like a typewriter except that it automatically prevents words from being chopped in half by the end of the line.

Programs 2 and 3 are not specifically for the printer, but the graphics they produce on the screen are quite effective when **COPY** is used. No. 2 takes up to 40 characters and enlarges them 4 times using **PLOT** and the ZX81's built-in character set. This program runs very slowly! Two examples of pretty pictures are plotted by the 2 programs given as No. 3, which may be easily modified to give other variations.

Both these last two programs plot rather slowly due to the amount of calculation they have to do, so program 4 is designed to tack on the end of either of them (or any graphics program). It saves the final picture in A\$: once this is done, the results of your program can be recalled quickly with **PRINT A\$**. The original program may then be erased if you wish – line by line, though; remember that **NEW** will clear the array you've just spent so much effort in filling up. A RAM pack will be necessary in order to fit in all the storage.

Program 5 is simple yet it enables complex pictures to be drawn on the screen and to be subsequently printed.

If key 1 is pressed a flashing black square will appear. This can be moved over the whole screen area using keys 5, 6, 7 and 8. It will leave a black trail. If key 0 is pressed the flashing black square will rub out everything in its path, as it is moved around the screen.

If key 2 is pressed the flashing square disappears and may be moved aside without upsetting your composition. When your picture is completed key 3 should be pressed and the picture will be printed out.

This program really needs the 16K RAM pack as only the most elementary picture can be drawn without it.

Combine this program with program 4 which stores the display so that your space monster is available for instant recall.

HIGH RESOLUTION PRINTER PROGRAM

This program works rather like **PLOT**, except that it works only with the printer, and uses a square of 256×256 points to give 4 times better resolution.

You will need a 16K RAM pack to run this program. Before entering it, type:

```
POKE 16389, 124  
NEW
```

to reserve some memory. If you don't, the program warns you and won't run. Be careful to enter line 1 exactly as given: the **RNDs** and **TAN** must be entered using function mode, not spelt out. Note that there is a space after the **█** symbol. The four full stops will have other characters; **POKE** into them after the first run of the program.

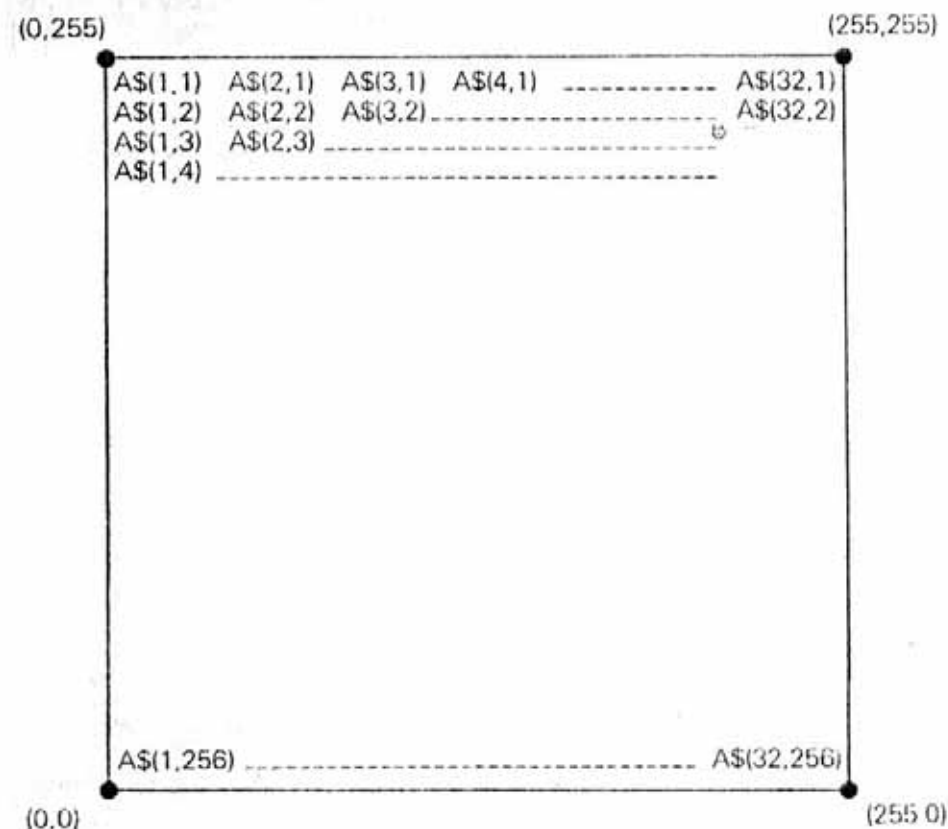
Two examples are given, differing only in the lines between 12 and 9960. No. 6 plots a "flower" shape. No. 7 plots a sine and cosine graph complete with axes. (Compare with the one you get using **PLOT** in Chapter 18 of the ZX81 books). The computer should be put in **FAST** mode before running the

program. Even so, there will be quite a wait before anything happens, and a further 4 minutes before printing is finished. Remember that the program has $256 \times 256 = 65536$ possible points to plot at.

PROGRAM DESCRIPTION

Lines 1 to 12 set up two distinct machine code subroutines: one at 16514 (the **REM** statement in line 1) the other at 31744 in reserved memory. The first simply **ORs** together two 8 bit numbers placed in 16526 and 16527, and allows more than one graph to be plotted on top of another. The second is the **LPRINT** routine, taken from the **ROM** and altered to use the data poked into the 256 bytes from 32256 onwards. Array A\$ takes 8K bytes of memory and contains the points to be plotted. Subroutine 9980 plots point (X,Y) into the array, and lines 9968 onwards take the finished array 256 elements at a time and hand it over to the print routine.

POSITION OF ARRAY ELEMENTS ON PRINT OUT



Each element of the array, e.g. A\$(1,256) contains one byte of information. A byte has eight bits which have the following values:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Each bit in each byte in the array represents one dot on the high resolution printout. Hence each element of the array represents eight successive dots, the left-most dot corresponding to bit 7 whose value is 2^7 and the right-most dot bit 0.

Hence:

Dot (0,0) corresponds to bit 7 of Array Element A\$(1,256)

Dot (1,0) corresponds to bit 6 of Array Element A\$(1,256)

Dot (255,255) corresponds to bit 0 of Array Element A\$(32,1)

Hence Dots (0,0) through to (7,0) are represented by bits 7 through to 0 of Array Element A\$(1,256). If you wish to print all of those dots, then use the command:

LET A\$(1,256)=CHR\$(27+2**6+2**5+2**4+2**3+2**2+2**1+1)**

If you only wish to print the first three dots then use the command.

LET A\$(1,256)=CHR\$(27+2**6+2**5)**

PROGRAM 1

```
5 REM TEXT EDIT
10 FAST
15 INPUT A$
20 IF LEN A$>32 THEN GOTO 35
25 LPRINT A$
30 GOTO 15
35 LET C=0
40 IF C=32 THEN GOTO 65
45 LET B$=A$(32-C)
50 IF A$(33-C)=" " OR B$="." OR B$="'" OR B$=":" OR
B$=";" OR B$="?" OR B$=CHR$ 11 THEN GOTO 70
55 LET C=C+1
60 GOTO 40
65 LET C=0
70 LPRINT A$(TO 32-C)
75 LET A$=A$(33-C TO)
80 IF A$(1)=" " THEN LET A$=A$(2 TO)
85 GOTO 20
```

THIS IS A SAMPLE OF TEXT AS IT MIGHT APPEAR BOTH BEFORE AND AFTER EDITING. IT IS TO BE HOPED THAT THAT HAS BEEN EDITED IS MORE INTELLIGIBLE THAN THAT THAT HAS NOT. HOWEVER WHAT DO YOU THINK? REMEMBER BEFORE YOU ANSWER THAT COMPUTERS HAVE FEELINGS TOO AND THAT A SULKING ZX81 CAN GET ITS OWN BACK IN THE MOST SUBTLE OF WAYS.

THIS IS A SAMPLE OF TEXT AS IT MIGHT APPEAR BOTH BEFORE AND AFTER EDITING. IT IS TO BE HOPED THAT THAT HAS BEEN EDITED IS MORE INTELLIGIBLE THAN THAT THAT HAS NOT. HOWEVER WHAT DO YOU THINK? REMEMBER BEFORE YOU ANSWER THAT COMPUTERS HAVE FEELINGS TOO AND THAT A SULKING ZX81 CAN GET ITS OWN BACK IN THE MOST SUBTLE OF WAYS.

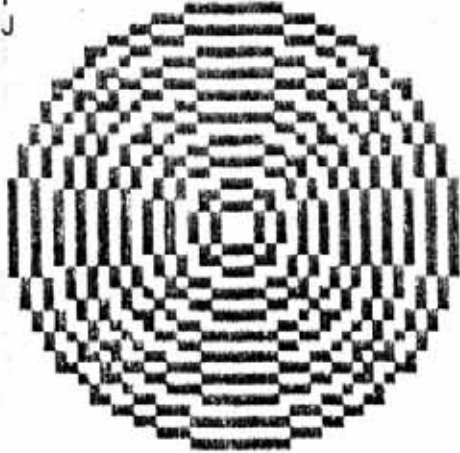
PROGRAM 2

```
10 DIM A$(40)
20 INPUT A$
30 FOR Y=43 TO 4 STEP -1
40 FOR X=0 TO 7
50 LET S=43-Y
60 LET N=X+8*INT (S/8)
70 LET SCAN=S-8*INT (S/8)
80 LET P=PEEK (7680+8*CODE A$(N+1)+SCAN)
90 FOR I=7 TO 0 STEP -1
100 PLOT 8*X+7-I,Y
110 UNPLOT 8*X+7-I,Y
120 IF P<2*I THEN GOTO 150
130 LET P=P-2*I
140 PLOT 8*X+7-I,Y
150 NEXT I
160 NEXT X
170 NEXT Y
```

SLOWEST
PROGRAM
IN THE
WORLD?

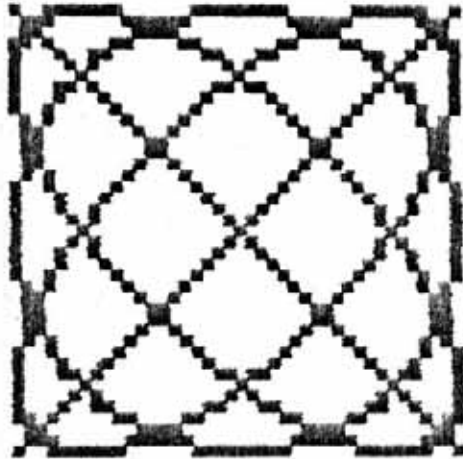
PROGRAM 3

```
10 FOR J=1 TO 10
20 FOR I=0 TO J*12
30 PLOT 32+J*2*SIN (I/(J*6)*PI),22+J*2*COS (I/(J*6)*PI)
40 NEXT I
50 NEXT J
```



PROGRAM 3a

```
10 FOR I=-4 TO 4
20 FOR J=0 TO 120
30 PLOT 32+20*SIN (J/60*PI),22+20*COS (J/60*PI+I/4*PI)
40 NEXT J
50 NEXT I
```



PROGRAM 4

```
1000 DIM AS(704)
1010 FOR I=0 TO 21
1020 FOR J=1 TO 32
1030 LET AS(J+32*I)=CHR$ PEEK (PEEK 16396+256*PEEK
16397+J+33*I)
1040 NEXT J
1050 NEXT I
```

PROGRAM 5

```

10 REM "SKETCHPAD"
20 LET X=32
30 LET Y=22
40 GOSUB 500
50 UNPLOT X,Y
60 PLOT X,Y
70 GOTO 40
80 GOSUB 500
90 GOTO 80
100 GOSUB 500
110 PLOT X,Y
120 UNPLOT X,Y
130 GOTO 100
500 IF INKEY$="1" THEN GOTO 40
510 IF INKEY$="2" THEN GOTO 80
520 IF INKEY$="3" THEN GOTO 600
530 IF INKEY$="0" THEN GOTO 100
540 IF INKEY$="5" THEN LET X=X-1
550 IF INKEY$="6" THEN LET Y=Y-1
555 IF Y=43 THEN GOTO 565
560 IF INKEY$="7" THEN LET Y=Y+1
565 IF X=63 THEN GOTO 590
570 IF INKEY$="8" THEN LET X=X+1
590 RETURN
600 COPY

```



PROGRAM 6

```

1 REM U$RND U$RND$ TAN
2 IF PEEK 16388+256*PEEK 16389=31744 THEN GOTO 5
3 PRINT "MEMORY NOT RESERVED"
4 STOP
5 FOR I=0 TO 112
6 POKE 31744+I,PEEK (2161+I)
7 NEXT I
8 POKE 31800,63
9 POKE 31857,201
10 POKE 16517,95
11 POKE 16524,79
12 DIM A$(32,256)
30 FOR I=0 TO 240
40 LET X=128+30*SIN (I/60*PI)
50 LET Y=128+120*COS (I/120*PI)
60 GOSUB 9980
70 LET X=128+120*COS (I/120*PI)
80 LET Y=128+30*SIN (I/60*PI)
90 GOSUB 9980
100 LET X=128+40*SIN (I/120*PI)
110 LET Y=128+40*COS (I/120*PI)
120 GOSUB 9980
130 LET X=128+80*SIN (I/120*PI)+20*SIN (I/60*PI)
140 LET Y=128+80*SIN (I/120*PI)-20*SIN (I/60*PI)
150 GOSUB 9980
160 LET X=128+30*SIN (I/120*PI)-20*SIN (I/60*PI)
170 LET Y=128-80*SIN (I/120*PI)+20*SIN (I/60*PI)
180 GOSUB 9980
190 NEXT I
1000 GOTO 9988
9980 REM PLOTS (X,Y) INTO A$
9981 IF X<0 OR X>255 OR Y<0 OR Y>255 THEN RETURN
9982 LET C=1+INT (X/8)
9983 LET R=256-INT Y
9984 POKE 16526,CODE A$(C,R)
9985 POKE 16527,2**(8*C-INT X-1)
9986 LET A$(C,R)=CHR$ (USR 16514)
9987 RETURN
9988 REM PRINTS A$ 8 LINES AT A TIME
9989 FOR I=0 TO 246 STEP 8
9990 FOR J=1 TO 32
9991 FOR K=1 TO 8

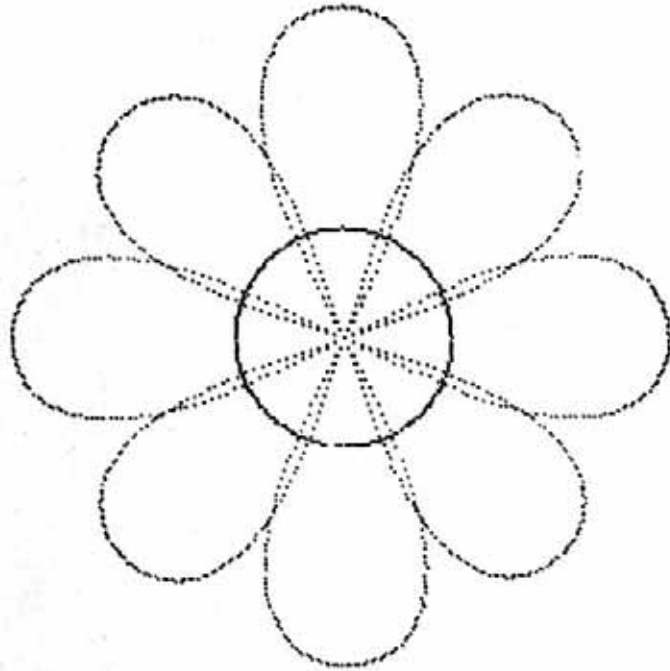
```


PROGRAM 6 (continued)

```

9992 POKE 32255+K+8*(J-1),CODE A$(J,K+I)
9993 NEXT K
9994 NEXT J
9995 FOR H=0 TO 31
9996 POKE 16444+H,H
9997 NEXT H
9998 LET HPRINT=USR 31744
9999 NEXT I

```



PROGRAM 7

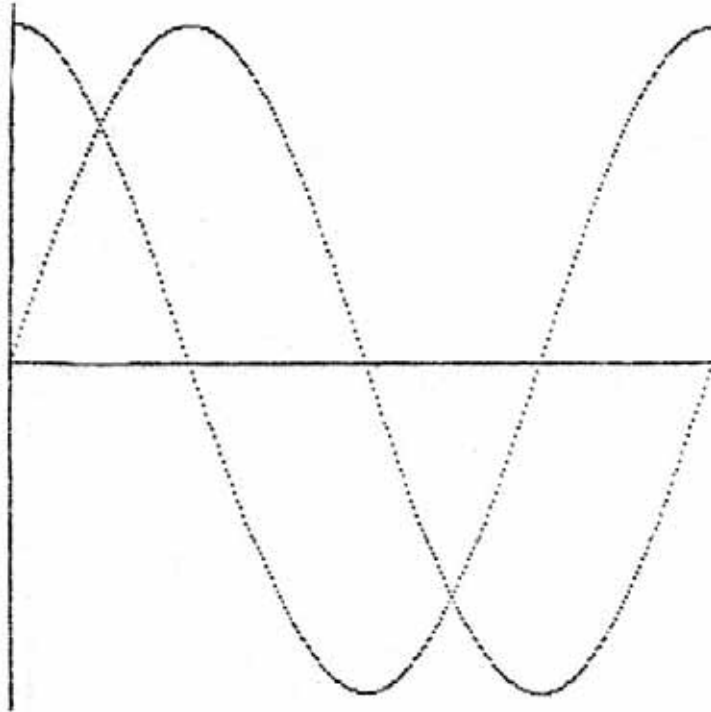
```

1 REM USRND USRND TAN . .
2 IF PEEK 16388+256*PEEK 16389=31744 THEN GOTO 5
3 PRINT "MEMORY NOT RESERVED"
4 STOP
5 FOR I=0 TO 112
6 POKE 31744+I,PEEK (2161+I)
7 NEXT I
8 POKE 31800,63
9 POKE 31857,201
10 POKE 16517,95
11 POKE 16524,79
12 DIM A$(32,256)
20 FOR I=1 TO 32
30 LET A$(I,128)=CHR$ 255
40 NEXT I
50 FOR I=1 TO 256
60 LET A$(1,I)="█"
70 NEXT I
75 LET A$(1,128)=CHR$ 255
80 FOR X=0 TO 255
90 LET Y=128+120*SIN (X/128*PI)
100 GOSUB 9980
110 LET Y=128+120*COS (X/128*PI)
120 GOSUB 9980
130 NEXT X
140 GOTO 9988
9980 REM PLOTS (X,Y) INTO A$
9981 IF X<0 OR X>255 OR Y<0 OR Y>255 THEN RETURN
9982 LET C=1+INT (X/8)
9983 LET R=256-INT Y
9984 POKE 16526,CODE A$(C,R)
9985 POKE 16527,2*(8*C-INT X-1)
9986 LET A$(C,R)=CHR$ (USR 16514)
9987 RETURN
9988 REM PRINTS A$ 8 LINES AT A TIME
9989 FOR I=0 TO 246 STEP 8
9990 FOR J=1 TO 32
9991 FOR K=1 TO 8
9992 POKE 32255+K+8*(J-1),CODE A$(J,K+I)
9993 NEXT K
9994 NEXT J
9995 FOR H=0 TO 31

```

PROGRAM 7 (continued)

```
9996 POKE 16444+H,H
9997 NEXT H
9998 LET HPRINT=USR 31744
9999 NEXT I
```



LPRINT

A printing error occurs when using **LPRINT** on decimal numbers in the range .00001 to .0099999999.

The leading zeroes, except the first, print as various non-numeric characters instead of zero.

This is overcome by converting the number you want to **LPRINT** into a string. Although the string looks the same, it will print correctly because the routines inside the computer treat strings and numbers differently. Alternatively **PRINT** on the TV screen then use **COPY**.

Suggested programming alternatives

Example 1

```
Replace: 100 LPRINT A
With:    100 LET A$=STR$ A
         110 LPRINT A$
```

Example 2

```
Replace: 100 LPRINT A+B/C-D, SIN (1-Q+N)
With:    100 LET A$=STR$ (A+B/C-D)
         110 LET B$=STR$ (SIN (1-Q+N))
         120 LPRINT A$, B$
```

Example 3

```
Replace: 100 LPRINT "VOLUME="; A*B*C; "LITRES"
With:    100 LET A$=STR$ (A*B*C)
         110 LPRINT "VOLUME="; A$; "LITRES"
```

Note

You only need to change your program if you are likely to **LPRINT** decimal numbers which would otherwise print incorrectly. This error is restricted to **LPRINT**, both **COPY** and **LLIST** work correctly.

GUARANTEE AND RETURNS INFORMATION
(see back cover)