

Vol 2 - No 9.

May 1989.

FORMAT

THE MONTHLY MAGAZINE FOR
SPECTRUM, DISCIPLE & PLUS D USERS



LOGICALLY
The Best For Your Spectrum

Vol 2 No 9. **CONTENTS** May 1989.

| | |
|------------------------------|----|
| The Editor Speaks..... | 3 |
| News On 4..... | 4 |
| An Odd Program..... | 5 |
| Secrets of Word Manager..... | 7 |
| • Inside G+DOS - Part 2..... | 11 |
| • Short Spot..... | 14 |
| Mandelbrot Pics..... | 15 |
| • Total Recall..... | 19 |
| The Hack Zone..... | 21 |
| Label Printer..... | 22 |
| The Adventure Corner..... | 23 |
| Beyond Simple UDGs Part 2... | 26 |
| Back Issues..... | 28 |
| FORTH Introduction..... | 29 |
| Small Ads..... | 31 |

THIS MONTHS ADVERTISERS:-

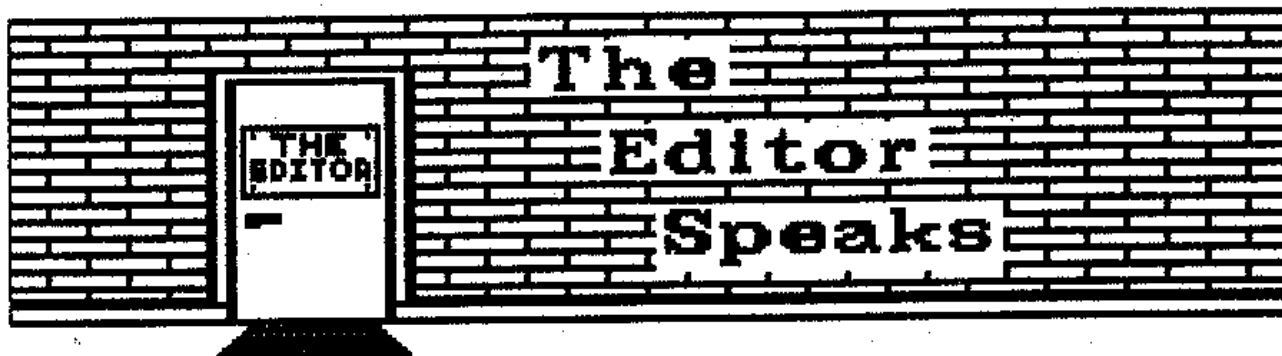
| | |
|------------------|------------|
| BETTERBYTES | Back Cover |
| BRADWAY SOFTWARE | 20 |
| KEMSOFT | 25 |
| LERM | 6 |
| P.C.G. | 10 |

(C)Copyright 1989 INDUG. All Rights Reserved.

No part of this publication may be reproduced, in any form, without the written consent of the publisher. FORMAT readers may copy program material only for their own personal use.

FORMAT is published by INDUG. 34 Bourton Road, Gloucester, GL4 0LE, England. Telephone 0452-412572. DISCIPLE and PLUS D are trade marks of MILES GORDON TECHNOLOGY, Lakeside Technology Park, Phoenix Way, Swansea, South Wales, SA7 9EH. Telephone 0792-791100. The DISCIPLE is marketed by Rockfort Products, 81 Church Road, Hendon, London, NW4 4DP.

Printed by D.S.LITHO. Gloucester. Tel:- (0452) 23198



The new look to FORMAT's pages seems to have gone down very well with our readers. Most people who I've spoken to liked the new column layout and were pleased with the extra content this enables me to print on a page. One or two complaints though. Mostly related to the new type face causing confusion in listings, the lower case letter L (1) does look almost identical to the numeric one (1). This has always been a problem with computer publications but I will try to make sure that upper case letters are used in future.

As if I am not overworked enough at the moment Your Sinclair magazine has caused a major problem for me. As long-standing readers will know we used to publish a small range of software under the heading FORMAT SOFTWARE SERVICE. The last advert for this appeared in September last year and I did make it clear in the December issue that the service had been suspended until further notice. Much to my surprise there were still a handful of orders arriving each week (proves some people skip my editorials) which wasted time in having to return them with an explanation. However YS really dropped me in it when they published a review of PLUS D HACKER in their April issue. I didn't send them a copy, no-one at Your Sinclair contacted me about the review (I can't understand the comment at the end of the review where they thank INDUG for its help) and they gave the UK 'Members Only' price.

Needless to say (as the review was a very good one) the orders have come flooding in. My course of action has been to forward them to the author -

Steve Nutting - who is making valiant efforts to sort thing out for customers as soon as he can. Meanwhile if you want a copy of PLUS D HACKER then see Steve's advert on the SMALL ADS page.

I'm sorry to say that as you read through FORMAT you will find several writers this month make the same comment - why do they never hear from readers? - come on, support our writers, or they might not be able to continue.

I have finally got round to drawing the winners for the Christmas Wordsquare Competition and If you have been lucky enough to win a copy of TRADE SECRETS by Geoff Bobker you will find it enclosed with this months issue. A full list of winners will be published next month.

This months issue has several features on computer languages including FORTH and Machine Code. Basic is of course well covered as usual. If you are using any other language I would like to hear from you as it would be interesting to see just how many languages there are available for the Spectrum.

At one time there were more books published for the Spectrum than for most other home computers put together. Where have they all gone? The Spectrum is still selling. Software for the Spectrum still sells more than any other micro. So where are the books? Anyone got an answer?

See you next month.

Bob Brenchley. Editor.

NEWS ON 4

APRIL FOOL BACKFIRES.

CRASH magazine went well beyond the bounds of good journalism with this years April Fools Spoof news item under the heading "Amstrad's New Hardware". In this they list several new hardware items Amstrad were about to launch including several add-ons for the Spectrum. They went on to quote MGT's Alan Miles as saying "Customers are likely to be dubious about the quality of the Amstrad hardware".

Alan Miles was never contacted by Crash, he never made such a remark to Crash nor to any other magazine for that matter.

Crash have promised to print a retraction in the next issue but the ill will that this sort of article can cause is difficult to erase just by printing an apology and retraction. There is a fine dividing line between good and bad taste. Crash did not exercise good taste in editing this article.

LIFE TIME DISC DRIVES.

As we announced last month MGT have now launched their 3.5" LIFE TIME DRIVES. This new concept in disc drives means you wont need to change your L.T.D. if you down-grade from a Spectrum to an Atari ST or IBM clone.

Each drive comes complete. Cased, with dip switches available without dissecting the drive, it has a built in power supply and the box includes a dust cover and technical manual for £129.95. All you then need is a cable for you computer - IBM or Amiga cost £18.50 while Spectrum, BBC and QL cables cost £10.

A 5.25" drive will be available in

the near future along with similar drives without power supplies.

The drives were officially launched at the European Trade Show in London where generated a lot of interest. Retailers like the idea of stocking one drive (plus a collection of cables) instead of the several different drives.

Full details are available from MGT on 0792-791100.

T.V. PRICE WAR.

A high street price war may be good news for any of our readers who would like a new television to go with their computer. It started when Liverpool based COLORVISION started selling 14inch colour TVs for just £99.99 in their 55 shops. Then the national giant Dixons slashed £30 of their own-brand set. Now others look set to follow.

While this may be good news in the short run it may lead to problems later on as it is bound to force many smaller traders out of business. A spokesman for the Radio Electrical and Television Retailers Association is quoted as saying that he could not understand how manufactures could make TVs at that price.

BBC BIAS

The BBC have shown blatant bias against Britain's No.1 Micro. On there long (total 2hrs 40mins) SOFTWARE show on Sunday April 2nd one micro was conspicuous by its absence. Yes readers a whole show and not one glimpse of the Spectrum. How about flooding the BBC with photos of the Spectrum so they know what one looks like before they organize another program?

An Odd Program...

By: Ray Elder.

A few weeks ago my eldest brought home a piece of work related to her coursework for the new GCSE exam. As I was bored at the time and also felt that I should show a parental interest, I asked her what she was doing. This program is a result of that coursework.

THE TASK.

The subject was maths (not my most favourite) and the work was as follows:-

Take any times table and calculate each step, if the result is greater than 9 (ie. two or more digits) then add the digits together and repeat this operation until the product is a single digit (ie. less than 10)

For example :-

5 times 1 = 5 ... Product = 5
5 times 2 = 10 .. Product = 1 (1+0)
5 times 3 = 15 .. Product = 6 (1+5)
etc.
5 times 23 = 115. Product = 7 (1+1+5)
etc.

Now, using graph paper, draw a line vertically up for the length of the first product, then for each other product draw a line at 90 degrees clockwise to the end of the previous line of length equal to the product.

So for our 5 times table example the first line is 5 units upwards, the next line 1 unit long going to the right, followed by a line of 6 units downwards followed by a line of 2 units going to the left.....etc.

How many different patterns can be formed?

There was no limit to the number of tables to be used and at first sight I thought she had a piece of work that would take a lifetime, I could imagine her at 60 still working out 23897 times 1, 23897 times 2,

So immediately I thought that a program to do the calculations would be useful, even using a calculator would mean entering each sum one at a time - sounds like a job for a computer!

The program was duly written, now it only took ages to transfer the results to graph form.

So I took on myself the task of utilising the computers graphics capability and low and behold, this program and, for me, the surprising results obtained. Try it yourself, curious isn't it!

A LITTLE NOTE.

Due to the computer screen pixels not being square there may be some slight distortion of the images/patterns, this is best seen by working the 9 times table. The result for this should be a perfect square, if not then you may have to put some form of scaling on the x or y lines to suit.

The program is written in a straightforward manner and it should be easy to add any refinements of your own such as screen print routines etc.

THE PROGRAM

```
10 CLS : INPUT "Which table to calculate? ";table
20 LET flag=1: LET x=90: LET y=60: PLOT x,y
```

```

30 LET limit=40
40 PRINT "Do you want step mode Y/N?"
   ": GOSUB 2000
50 LET mode=1: IF k$="y" THEN LET mode=0
60 LET counter=1
70 LET a$=STR$(table*counter)
80 LET product=0
90 FOR i=1 TO LEN a$
100 LET product=product+VAL a$(i)
110 NEXT i
120 IF product>9 THEN LET a$=STR$(product)
    GOTO 80
130 PRINT AT 0,0;table;" times "
    ;counter;"=";table*counter;"
    ... Product="
    ;product;" "
150 GOSUB 1000
160 IF mode=0 THEN
    PAUSE 1000
170 LET counter=counter+1
180 IF counter<=limit THEN GOTO 70
190 PRINT AT 21,0;
    "Run program again Y/N?"
200 GOSUB 2000
210 IF k$="y" THEN
    RUN
220 STOP
1000 LET scale=product*5
1010 IF flag=1 THEN
    DRAW 0,scale
1020 IF flag=2 THEN
    DRAW scale,0
1030 IF flag=3 THEN
    DRAW 0,-scale
1040 IF flag=4 THEN
    DRAW -scale,0
1050 LET flag=flag+1: IF flag=5 THEN LET flag=1
1060 RETURN
2000 LET k$=INKEY$:
    IF k$<>"y" AND k$<>"n" THEN
        GOTO 2000
2010 RETURN

```

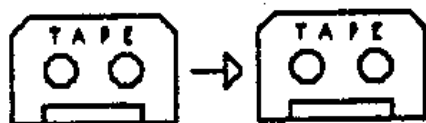
Now its your turn. Try altering the maximum product allowed, thats the

test in line 120, to a different value and see what you get with the same tables. You may need to alter the limit (see line 30) so that a full repeat of the pattern is drawn.

Mathematical patterns are interesting, there are many more for you to explore. If you find any really good ones then send them in to FORMAT.

* - * - * - *

SPECTRUM SOFTWARE by LERM. WITH FULL MONEY BACK GUARANTEE.



TAPE UTILITY 1

TU.1 - COPIES MOST TAPES EVEN JERKY AND MANY OF THE RECENT COUNTDOWN LOADERS. EVEN CONVERTS MANY SECURED PROGRAMS TO NORMAL SPEED FOR RELIABLE LOADING + TRANSFER. MEASURES LOADING SPEED. COPIES EXACTLY BLOCKS - EVEN OVER 51K. ETC. £7.99



MICROMATE

"PERFECT ADDITION TO M/DRIVE"
"MORE POWER WITHOUT THE PRICE"
15 JULY 1988

THE PROGRAM PAYS FOR ITSELF. FOR 1/2 DRIVE SYSTEMS. EVEN IF YOU HAVE A M/DRIVE UTILITY YOU HAVEN'T ANYTHING LIKE MICROMATE. OPTIONS INCLUDE: FORMAT (APPROX 104K), CLONE, SECTOR EDITOR, BOOT PROG, RENAME, RECONDITIONS AND REPAIRS CART, MULTIPLE ERASE, TRUN CAT, TAPE TO DRIVE, UNIQUE FAST DRIVE TO DRIVE COPIER, DRIVE TO TAPE (+ NONE STOP...). SOLD ON CARTRIDGE - NORMALLY £13.99. SPECIAL OFFER £11.99 - LIMITED PERIOD ONLY.

BRAND NEW M/DRIVE CARTRIDGES - £1.75 EACH POST FREE.
CODE COMPRESSOR - £4.50. UPDATE SERVICE AVAILABLE.

STATE TYPE OF SPECTRUM + DISC SYSTEM WHEN ORDERING
*.S PLUS 3 or 128K Sinclair with OPUS.

SAE FOR DETAILS. POST FREE BUT +£1 Europe, £2 rest.



TPS.

TPS - TRANSFERS HUNDREDS OF PROGRAMS FROM TAPE TO YOUR DRIVE SYSTEM (INC PLUS 3). MANAGES SOME PROGRAMS HARDWARE DEVICES CAN'T DO! HANDLES JERKY AND COUNTDOWN LOADERS EASILY. INCLUDES USEFUL UTILITIES. NOT FOR THE NOVICE. £12 (+£1 ON M/DRIVE). INFORMATION BOOK 1 - DETAILS ON 400 TRANSFERS USING TPS £2.50. BOOK 2 COVERS 180 FOR £2.50 - MANY TOP 20 NITS INCLUDING SOME 128K PROGRAMS.



Z80

TOOLKIT 2.

"EXCELLENT VALUE FOR MONEY"
"BRILLIANT PACKAGE"

"IF ITS NOT IN TOOLKIT YOU DON'T NEED IT". 15 JULY 1988. POWERFUL ASSEMBLER, AND NOW A PROGRAM TO DO THE REVERSE - M/CODE TO SOURCE. TOOLKIT WITH DISASSEMBLER, SINGLE STEPPER, DE-BUG, HEX TO DEC. ETC. UNRIVALLED AT £8.99

LERM

DEPT. F, 11 BEACONSFIELD CLOSE,
WHITLEY BAY, TYNE AND WEAR NE25 9UW
☎ 091-253 3615

THE SECRETS OF WORD MANAGER

SPECTRUM MACHINE CODE MADE EASY

By: Francis Miles.

As the author of a long and complicated machine code program, **WORD MANAGER**, I have several times been asked "How did you learn machine code programming?", and found it difficult to give a sensible answer.

I believe there is a shortage of literature on machine code programming beyond the elementary stages; for people who have got some kind of grasp of how assembly language works, but have not yet begun to feel any real freedom in using it. I thought I could perhaps do something to fill this gap, by writing about some of the things I had to find out by trial and error, mostly error, in the long and adventurous process of writing my own word processor program. (It was almost the first machine code program I ever wrote: only a lunatic would attempt a 17K program as his first effort, but just ask anyone who knows me.)

These notes are not a description of **WORD MANAGER**, and you don't have to know the program to understand them (I hope). I have merely used **WORD MANAGER** as a quarry from which to excavate examples illustrating various techniques and programming practices. An ounce of example is worth a ton of philosophizing; I couldn't cut out all the philosophizing, but have tried to swamp it with examples.

My only other advice to people who want to try their hand at machine code programming would be:-

1. Get some reference books; I can recommend "The Complete Spectrum ROM Disassembly" by Drs Logan and O'Hara (Melbourne House, 1983) and "Z80 Instruction Handbook" by Nat Wadsworth (Hayden Book Co, Rochelle Park, New Jersey, 1978).

2. Get an editor/assembler program. You can write machine code directly byte by byte in machine language, but that way madness lies!

At the request of the editor I have avoided the use of HEXadecimal numbers which he believes is one of the greatest causes of confusion in newcomers to machine code. Where I mention a Spectrum ROM routine I do give its hex address so you can find it in the "The Complete Spectrum ROM Disassembly" mentioned above.

So now, as they say, on with the show.

PRINTING MESSAGES ON SCREEN. Part 1.

WORD MANAGER uses four different systems for putting messages on the screen.

System 1. The simplest is the "restart" subroutine RST 16 (RST 10H). This calls upon the ROM to print one byte from the A register at the current print position on the screen. Before you call this subroutine, you must open the appropriate channel:-

Channel 1 - to print in the "input area", the bottom two (sometimes more) lines on the screen.

Channel 2 - to print in the upper part of the screen.

This is simply done by using the Spectrum's own CH.OPEN routine at 5633 (1601hex):-

```
2420 A1C      EQU $
2430 ;Open channel 1.
2440         LD A,1
2450         JR A.C
2460
```

```

2470 A2C EQU $
2480 ;Open channel 2.
2490 LD A,2
2500 A.C CALL 5633 ;CH.OPEN
2510 RET

```

A call to A1C opens channel 1, to A2C opens channel 2. RST 16 is not very often used directly in **WORD MANAGER**: an example is in printing digital inputs of such things as "Number of copies required", in the input area. After channel 1 has been opened, and the required digit obtained in the C register, there follows:-

```

4220 LD A,C
4230 ADD A,48
4240 RST 16
4250 LD A,95
4260 RST 16
4270 LD A,8
4280 RST 16

```

The effect is to print the digit (the character code of any digit is 48 + the value of the digit), followed by the underline symbol code 95, which is used as a cursor, followed by a backspace code 8, so that the next digit input will overprint the underline symbol.

System 2. This uses the useful ROM subroutine at 8252 (203Ch) PR.STRING. Before you call this subroutine, you must:-

- 2.1 Open the appropriate channel, using subroutines A1C or A2C shown above.
- 2.2 Put the start address of the message you want to print in the DE register.
- 2.3 Put the length of the message (the number of characters to be printed) in the BC register.

This again is not used much in **WORD MANAGER**, having been largely replaced by the next two systems, but it is used for example to print the filename reminder at the bottom left of the menu screen. Whenever you LOAD, SAVE or MERGE with **WORD MANAGER** the filename that you input is copied to a 10-character block in the Spectrum's print buffer beginning at 23297, and

the length of the filename is recorded at 23362, also in the print buffer. So whenever the menu is printed out this subroutine is called:-

```

5620 P.FNAM EQU $
5630 ;Print filename.
5640 LD A,(23362)
5650 AND A
5660 RET Z
5670 CALL A1C
5680 LD DE,23297
5690 LD BC,(23362)
5700 CALL 8252 ;PR.STRING
5710 RET

```

Note lines 5640 to 5660: if the filename is blank its length will be zero, and the subroutine will return from line 5660. Terrible things happen if you try calling PR.STRING with zero in the BC register.

Another example is the printing of "Start tape and press any key" in **WORD MANAGER**'s cassette SAVE routine (which is not quite the same as the Spectrum's ROM routine). This message is already in ROM, at 2466 (09A2h), with its final fullstop marked as described in 3. below. You can look at it on screen if you have a copy of **WORD MANAGER** by running the following two-line BASIC program:-

```

1 REM Read m/c
2 FOR F=2466 TO 70000: PRINT F; TAB 7
; PEEK F,: IF PEEK F>32 THEN PRINT CH
R$ PEEK F;
3 PRINT: NEXT F

```

This simple little program is incredibly useful for all sorts of things.

For some reason which I now forget, I wanted to print this message without its final fullstop; this was simply achieved by:-

```

3340 LD DE,2466
3350 LD BC,30
3360 CALL 8252 ;PR.STRING

```

(A2C had been called already).

System 3. Another useful ROM routine is PO.MSG at 3082 (0COAh). To use this

you must have a string of messages in your machine code, up to 256 messages, and the end of each message must be marked by setting bit 7 of the final character of the message; equivalent to adding 128 to the character code. And the whole table must be prefixed with a 128 byte. There are examples in the ROM at 2465 (09A1h), 3320 (OCF8h) (the "scroll?" message - I don't know why the programmers put this one by itself), and 5009 (1391h) (the "error reports"). You can read them by using the "Read m/c" routine given above. Before you call 3082 PO.MSG you must also:-

- 3.1 Call A1C or A2C as usual.
- 3.2 Put the serial number of your message (0 for the first message in the table, 1 for the second, and so on) in the A register.
- 3.3 Put the start address of the table in the DE register.

In **WORD MANAGER** there is only one message string, whose start is labelled M.TAB; parts of it look like this:-

```

0050 M.TAB DEFB 128
0060 ;0
0070 ;AT 2,0 PAPER 0 INK 9 [see Note
0080 DEFB 22,2,0,17,0,16,9
0090
0100 DEFM "0 for BLACK "
0110 DEFB 13 ;13 is <CR>
0120 DEFB 17,1 ;PAPER 1
0125 DEFB 16,9 ;INK 9
0130 DEFM "1 for BLUE "
... etc

0330 DEFM "ENTER to "
0335 DEFM "return to"
0340 DEFB 13
0350 DEFM "script displa"
0360 DEFB "y"+128
0370
0380 ;1
0390 DEFB "o","n"+128
0400
0410 ;2
0420 DEFB "o","f","f"+128
0430
0440 ;3
...
3020 ;31
3120 DEFM "Input key word:"

```

```

3130 DEFB " "+128
...

```

Note: In case you aren't familiar with the "embedded" control codes, which are invaluable for printing on screen by any system, they are:-

```

16,x INK x
17,x PAPER x
18,0/1 FLASH 0/1
19,0/1 BRIGHT 0/1
20,0/1 INVERSE 0/1
21,0/1 OVER 0/1
22,x,y AT x,y
23,x,0 TAB x
23,x,y TAB (x + 256y)

```

The messages from the list are printed by any of four subroutines which look like this:-

```

5730 P.SCOL EQU $
5740 ;Clear screen, with colours as
5745 ;in A, and print message
5750 ;serial number B.
5760 PUSH BC
5765 ;SCOL is another subroutine,
5770 ;which sets the screen colours
5775 ;and also calls A2C)
5780 CALL SCOL
5790 JR P.MES2
5800
5810 P.MES1 EQU $
5820 ;Print mssg A in input area.
5830 PUSH AF
5840 CALL A1C
5850 JR P.MES2
5860
5870 P.MESC EQU $
5880 ;CLS and print message number A
5890 PUSH AF
5900 CALL CLS;clears screen
5910 POP AF
5920
5930 P.MESS EQU $
5940 ;Print mssg A on screen.
5950 PUSH AF
5960 CALL A2C
5970 P.MES2 POP AF
5980 LD DE,M.TAB
5990 CALL 3082 ;PO.MSG
6000 CALL A2C
6010 RET

```

The vital line is 5990, which all these subroutines reach (after various preliminaries) with M.TAB, the start

of the message string, in DE and the message serial number in A. Notice that line 6000 reopens channel 2; some ROM printout routines cancel the opening of the channel (though not any of those described here). Here and there in **WORD MANAGER** the printing of a message is followed by an RST 16 or printing another message, and it saves a few bytes of memory to have the reopening in the subroutines. A simple example of the use of this system is the printing at the top of the screen of message no 31, "Input key word: ", when the user is being invited to find his place in the text:-

```

1430      LD A,(COLS) ;a single
1440 ;byte signalling the required
1445 ;PAPER and INK
1450      LD B,31 ;message No.
1460      CALL P.SCOL

```

A slightly less simple example shows how "Automatic justification off" or "on" is printed in the Display Control menu. The first two words "Automatic justification ", with their following

space, are printed using system 4., as described below. Then the program reads one of its flag bytes, MISFL; each of the 8 bits of this byte carries a different message, and the message of bit 5 is:-

```

bit 5 is 1 (set) = auto justification
is off,
bit 5 is 0 (reset) = auto
justification is on.

```

Depending on the state of this byte the program prints message 1 ("on") or message 2 ("off").

```

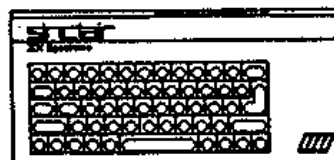
0230      LD A,(MISFL)
0240      RES 1,A
0245      LD (MISFL),A
0250 ;bit 1 reset for later op.
0260      BIT 5,A ;on or off?
0270      LD A,1 ;print "on"
0280      JR Z,AJ.PJ
0290      INC A ;print "off"
0300      AJ.PJ CALL P.MESS

```

I will deal with System 4. (and one other) next month.

P.C.G.

61 School Street
Barrow-in-Furness
Cumbria
LA14 1EW



Software:

| | |
|---------------------------------------|--------|
| WordMaster word processor | £11.90 |
| Headliner graphic designer | £8.95 |
| TypeIner desktop publisher | £16.95 |
| DTP Pack (all three above programs) | £37.80 |
| Devpac machine-code assembler | £15.95 |
| HiSoft BASIC floating point compiler | £24.95 |
| HiSoft C language system | £25.00 |
| TasSign sign designer | £17.95 |
| TasCalc spreadsheet | £17.95 |
| CP/M Plus operating system for the +3 | £29.95 |
| Masterfile +3 database | £25.00 |
| TasWord +3 word processor | £19.95 |
| TasSpell +3 spell checker | £19.95 |
| Stocks & Shares manager | £14.95 |
| Coursemaster horse-racing tipster | £14.95 |

Hardware:

| | |
|--|---------|
| Plus D disk & printer interface | £59.95 |
| Plus D with 3.5" disk drive | £159.95 |
| Plus D systems come with FREE printer cable! | |

If you're interested in
Programming,
Word Processing,
Business Software,
Desk-Top Publishing,
or just plain sick of games,
contact PCG for details of
these amazing programs.

Phone 0229-36957 now or
send an SAE for catalogue.

We have software for OPUS,
Disciple, Plus D, microdrive, +3
and many other systems. Call
now for details & prices.

Desk-Top Publishing:

Send now for details of the
amazing DTP Pack!

INSIDE

G+DOS

PART 2.

By: Stephen Warr.

As I mentioned last month, there are four addresses at which the PLUS D pages itself in. These are 0, 8, 58 and 102.

Starting with the easiest to explain, location 0 is the first address that the ROM comes to when you switch on the computer or press the reset button. The PLUS D pages itself in at this address so that it can initialise the drives before jumping to the normal memory reset routine.

Address 8 is the most important of the four addresses because the normal ROM comes here whenever an error occurs in BASIC. Usually the routine at location 8 directs the ROM to the current error handling routine which may either print a flashing question mark, if the line is being edited, or the relevant error message if the line is being run. However when the PLUS D pages in it starts looking for any of the extra commands that it allows, including the 'RUN' command used to boot your system file. The Command Codes and the extra streams available, eg. the centronics port and stream 'D' (the opentype files), are also directed to location 8.

NB. One of the few bugs in G+DOS ROM is that INPUT #3;a\$ will jump to an almost random address.

Paging in at location 58 is, perhaps, a bit surprising, as it occurs just after the beginning of the normal ROM's interrupt routine. When interrupts are enabled, the Z80 processor stops what it is doing every 1/50th of a second and calls an interrupt routine. In the Spectrum ROM this routine is used to scan the keyboard, but with a PLUS D attached the routine does two extra things as

well. Firstly it loads the data needed by the centronics port into the channel information area in case the bytes have been given their initial values by a system reset or a 'NEW'. Secondly it checks to see if the G+DOS RAM area contains anything at all. If not, it clears the RAM and gives the first 104 bytes their default values. This is why the PLUS D's red LED indicator flickers when you first switch on the power - it is the RAM being cleared as soon as the first interrupt occurs. POKE @7652,0 gives a quick way of resetting the whole system file using this method.

What all this simply means is that while the normal ROM is running, the PLUS D is still paging itself in and out 50 times a second! When the system file is loaded, the interrupt routine can also be vectored into RAM, but because G+DOS RAM is not reset by the normal memory reset routine, you can produce some strange and sometimes particularly unhelpful results if you are not careful. Try for instance:-

```
POKE @143,201: POKE @144,0:
POKE @145,0: POKE @143,195
```

The final paging address at location 102 has two purposes. Whenever you press the snapshot button, all that is really happening is CALL 102 (a Non Maskable Interrupt - NMI). Secondly, this address is also used as a return from a RST 16(#10) call - when G+DOS is paged in it is still possible to call a subroutine in the normal ROM. This is done in exactly the same way as Interface 1 did it. The instruction RST 16 is used followed by two bytes of data which give the address to call (RST 16 does the same as CALL 16 but it is only 1 byte long). The routine at location 16 in G+DOS then loads

15845 (#3DE5) with the value 71 (a capital G used to distinguish the call from pressing the snapshot button) before stacking 102 as a return address and making the call to the normal ROM. If you read that carefully it should all make sense.

That's quite enough technical information for now, let us move on to something a bit more interesting. On any disc formatted by the PLUS D or the DISCiPLE, the first 20K (40 sectors) are set aside to contain what is known as the DIRECTORY. For each file on disc there is one entry in the directory, up to a maximum of 80 files.

Access to the directory is probably the most important of all the disc routines. Before loading a file G+DOS must first search the directory to see if the file actually exists and then it must get the start address and length of the file before the file can be loaded. Saving is even more complex because as well as checking that the file doesn't exist, G+DOS needs to know which sectors are unused and can be saved to.

The routine that does all this, and more, is held at 2469 (#09A5). It is called with the machine code 'A' register holding a certain value depending on what you want to do:-

- A=1 Searches for a file that has a given filename.
- A=2 Prints an abbreviated catalogue to the current channel. Requires a filename.
- A=4 Prints a full catalogue to the current channel. Requires a filename.
- A=8 Searches for a file that has a given filename and is of a given file type.
- A=16 Searches for a file that has a given filename.
- A=32 Produces a disc map at #3A00. Explained in a later article.
- A=64 Exits as soon as an unused directory entry is found.

As you can see, most require at least one extra parameter.

A filename is the number that is printed in the left-most column when you display a full catalogue. When A=1, load the filename to 15874 (#3E02) before calling the directory routine.

A filename is a string of up to 10 characters which should be loaded to address 15878 (#3E06) upwards. Wildcards "?" and "*" can be used but unless "*" is the last character, the filename should be followed by spaces to make the length up to 10.

When A=8 the file type should be loaded to 15877 (#3E05). Its allowed values and their meanings are:-

- 1=BASIC program.
- 2=Numeric array.
- 3=String array.
- 4=CODE file.
- 5=48K Snapshot.
- 6=File produced using microdrive hook codes.
- 7=SCREEN\$ file.
- 8="Special" file.
- 9=128K Snapshot.
- 10=OPENTYPE file.
- 12=EXECUTE file.

The routine will use the current drive and returns with the zero flag set if the specified directory entry was found and with registers D & E holding the sector and track numbers respectively that the entry was found on. If the routine reached the end of the directory then the zero flag will be reset.

We will use the normal RAM for the moment, so assemble the routine below to 49152 for instance and then run it with RANDOMISE USR 49152:-

```

RST 8
DEFB 71      ;Page in G+DOS.
LD A,2       ;Use the main ROM routine
RST 16       ;to select channel 2 -
DEFW 5633    ;the upper screen.
LD HL,15878
LD (HL),"*"  ;Set filename.
LD A,4
CALL 2469    ;Produce full catalogue.
JP 80        ;Page out G+DOS
              ;and return.

```

As you can see, its effect is the same as typing **CAT 1** but without the first and last message lines. Of course, it is easier to do this from BASIC and so it is a little pointless, however we can also do several things that are impossible from BASIC and there are several routines in G+DOS to help us:-

2696 (#A88) Continues looking for further entries in the directory after a CALL 2469. Note - DE must not have been altered.

1853 (#73D) Calculates the filename of the directory entry from D & E and returns the result in the A register.

2452 (#994) Prints the filename of the current entry to the current channel.

6039 (#1797) Prints a space to the current channel.

6041 (#1799) Prints the character in the A register.

5958 (#1746) Prints as a decimal number the value held in HL. Preceding zeros (eg. 00001) are instead printed as the character held in the A register. NB. DE is altered.

5956 (#1744) As above but preceding zeros are printed as spaces.

5964 (#174C) As 5964 but only 4 characters are printed, ie. HL must be less than 10,000.

5970 (#1752) As 5958 but HL must be less than 1000.

5976 (#1758) As 5958 but HL must be less than 100.

Now for another practical example. The program below prints the filename and name of all the 48K snapshots on a disc:-

```
RST 8
DEFB 71
LD A,2
RST 16
DEFW 5633 ;Print to the screen
LD HL,15877
```

```
LD (HL),5 ;Set type = 48K snapshot
INC HL
LD (HL),"*" ;Set filename
LD A,8
CALL 2469
LOOP:JP NZ,80 ;Exit if no entries.
PUSH DE
CALL 1853
LD H,0
LD L,A ;HL=filename
LD A,32 ;Preceding zeros printed as
      spaces
CALL 5976 ;Print HL
CALL 6039
CALL 2452
LD A,13
CALL 6041 ;Print a newline
POP DE ;Restore track and sector
      numbers
CALL 2696 ;Search for further entries
JR LOOP
```

Next month I will be giving the details of what each directory entry contains, and there will be a routine to print an alphabetical catalogue. Until then, I would like to leave you with a few quirks of G+DOS.

If after typing in that really long program you suddenly find to your horror that you forgot to load your system file, DON'T PANIC, just type RUN 9999, or indeed any line number beyond the end of your program.

The POKE @ command, unlike the ordinary POKE, can actually poke a two byte value ie. a value between 0 and 65536, but numbers less than 256 are treated as one byte values.

Two quickies now, the OPEN # command can actually open a channel to any type of file not just an OPENTYPE file and the CAT command can also take the form CAT 1;"FILENAME".

The final piece of delectable information is that the SAVE d1"filename"CODE start,length command also allows an optional third parameter which is the address jumped to after the file is loaded back, allowing auto-running of machine code files.

More next month.

SHORT • SPOT

By: John Wase.

I was going to start with a Big Moan. Two months, and nothing suitable. In fact, I'd nearly written this article, when the short items started to trickle in. I haven't had time to try them all, but I've now been able to include some. I'm still short for next month, though, so please don't delay; send them in now. They needn't be complete programs, they certainly needn't be disc specific; they can even be hints, but they must do something useful!

Years ago, when I edited a Discovery column, the first things that came in were super-Directory/menu programs. In desperation, I've included mine as "filling": it's not original; Bob sorted out the solution to a problem I had. On the Microdrive (or Discovery, for that matter) doing NEW, then RUN and ENTER autoruns the program on disc called "run". So if you've crashed the program, reset, RUN and ENTER and you're away. Try it with the +D and you get that beastly error message "OK G-DOS 0:1" because the system file in the shadow ROM is still intact. A pain, isn't it. And worse if this is being used in School, where a menu comes up, then press keys 1-9 for the appropriate program. I mean, they're learning, and they've no idea how to start the thing up again, have they? Well, here's how to have a totally autorunning disc.

Save a program called autorun on your disc along these lines:-

```
1 REM autorun line 1 menu program
5 POKE @ 6000,10
10 PRINT "What program would you like?
   ": PRINT
20 PRINT " 1. Tasword": PRINT " 2. Mas
   terfile": PRINT " 3. Paperboy": PRI
   NT " 4. Utility"
50 LET a$= INKEY$
51 IF a$="1" THEN LOAD D1;"Tasword"
```

```
52 IF a$="2" THEN LOAD D1;"Masterfile"
53 IF a$="3" THEN LOAD D1;"Paperboy"
54 IF a$="4" THEN LOAD D1;"Utility"
90 BEEP 0.1,20: GOTO 50
```

When you run the boot disc, location 6000 is automatically read as part of a DOS checksum. If there is parity (i.e. if the system file is still intact) it doesn't load again. Poke 10 into it and you fool the thing into thinking DOS is not loaded. Good, eh!

Now for a quick tip. Did you know that until Bob's got his proper binders ready, 5.25" cardboard disc boxes keep A5 manuals (and FORMATS) tidy.

Harold Burton of Edinburgh wrote to complain that when he goes on holiday to somewhere warm (and overeats), all the weighing machines are calibrated in Kilos! By entering the appropriate range for you and yours, you can read the dial in familiar units and shock yourself without waiting to come home again!

```
10 REM Converter Kgs to St.lbs
20 PRINT AT 10,0;"KILOGRAMS-STONES/LB
   S CONVERTER"
30 INPUT "TO SCREEN? (Enter 2)""TO P
   RINTER? (Enter 3) ":p
40 CLS: PRINT #p; AT 1,0;"EQUIVALENT
   IN STONES & LBS"
50 INPUT "ENTER RANGE REQUIRED:"""FR
   OM Kg: ";a,"TO Kg: ";b
60 PRINT #p
70 FOR n=a TO b STEP 0.5
80 LET x=INT (2.2046*n/14)
90 LET y=INT ((2.2046*n/14-x)*14+0.5)
100 IF y=14 THEN LET y=0: LET x=x+1
110 PRINT #p;n;TAB 5;"Kg";TAB 9;"=";TA
   B (13 AND x>=10);x;" st";TAB (19 A
   ND y <10)+(18 AND y>=10);y;" lb"
120 NEXT n
```

See you again as soon as I've some bits and pieces. John Wase.

MANDELBROT PICTURES

By: Ray Williams.

In response to the editors appeal in for examples of Computer Art I thought readers would be interested in a few Mandelbrot set pictures.

The pictures were generated and printed on a 48K Spectrum system running a PASCAL program I call MnJMaps. The system includes a DISCiPLE, two 3.5" disc drives (MnJMaps uses only one of them) and a Panasonic KX-P1081 dot matrix printer. The program was edited and compiled by HiSoft's package "DISCiPLE Pascal 1.7" and runs under the run-time system that forms part of the package.

MnJMaps computes and stores on disc a selectable region of a Mandelbrot map, or of a Julia map, and provides facilities for displaying and printing a wide variety of representations of any previously stored map.

For each of the pictures, each dot position corresponds to a pixel position which in turn corresponds to a point $c = a + jb$ on the complex plane. Using the Mandelbrot set recurrence formula, the program computes a sequence of complex numbers $z(n)$ for each of these points until either the sequence is judged to be unbounded or the number of recurrence loops eventually traversed, which I call Loops, reaches a selectable large number (e.g. 1000) which I call MaxLoops. The recurrence formula is:-

$$z(n+1) = (z(n))^2 + c$$

where n is the infinite sequence of integers 0, 1, 2, 3, etc. The Mandelbrot set is the set of points c for which this sequence is bounded (i.e. the modulus of $z(n)$ remains finite) given that $z(0) = 0$. Regions yielding the most interesting pictures

lie just outside the perimeter of the set.



Map 1.

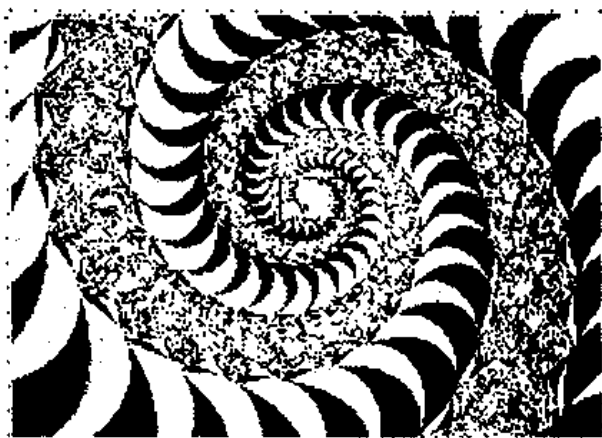
MnJMaps scans through the selected region pixel by pixel. For each pixel, it eventually stores on disc a byte "proportional to" the value of Loops. In this way the program gradually writes a number of files (currently 15) constituting a map of the selected region. These files are subsequently read to allow a representation of the map to be displayed and printed.



Map 2.

In the first picture, the black dots correspond to points for which Loops

is an even number less than 50 or for which Loops equals MaxLoops. Also printed is a rectangular window outlining the region of the next picture of a "zoom" series. MnJMaps includes a facility for adjusting the size and position of the window and for remembering these parameters for the next map in the series. Finally, the picture is surrounded by a "ruler": each division corresponds to 10 pixels and also to the Scale of the map.



Map 3.

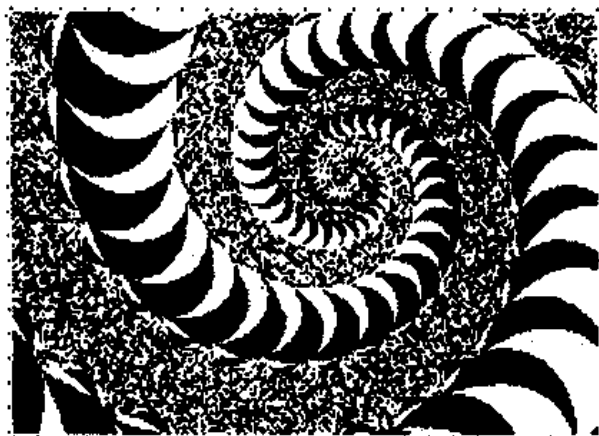
The picture was printed in two passes, one for the points that won an even number and one to print the window and the points that won MaxLoops. Before the first pass, I drew a mark on the paper so that I could wind the paper back to the same point for the second pass. (By changing the ribbon as well I could have printed the second pass in a different colour.) All the pictures were printed in this way except that in some cases I carelessly substituted odd for even in the first pass or did not show the window because the map was the last of a "zoom" series or because I forgot to show it.

The basis of the maps that the six pictures represent is tabulated here. R_1 and Im are the real and imaginary coordinates (a and b) of the origin (bottom left corner) of the map. **Scale** has the meaning defined above. MaxLoops was 1000 in every case. The Spectrum crunched numbers for about 6 hours to make the first map (which contains a fair number of points that

are in the set) and for about 4 hours for each of the others.

| <u>Map</u> | <u>R_1</u> | <u>Im</u> | <u>Scale</u> |
|------------|-------------------------|------------------------|--------------|
| 1 | -7.82000E-1 | 1.10000E-1 | 2.00000E-3 |
| 2 | -7.81600E-1 | 1.20200E-1 | 5.47619E-4 |
| 3 | -7.75412E-1 | 1.23650E-1 | 5.86734E-5 |
| 4 | -7.74772E-1 | 1.24166E-1 | 5.82078E-6 |
| 5 | -7.75138E-1 | 1.24690E-1 | 3.47694E-5 |
| 6 | -7.74533E-1 | 1.24941E-1 | 4.96706E-6 |

The six maps comprise two "zoom" series. The first series consists of maps 1, 2, 3 and 4. The second series consists of maps 1, 2, 5 and 6. On picture number 2 I forgot to show the second window outlining map number 5: this could be added by hand using the "ruler" on picture number 2 and the data tabulated above.



Map 4.

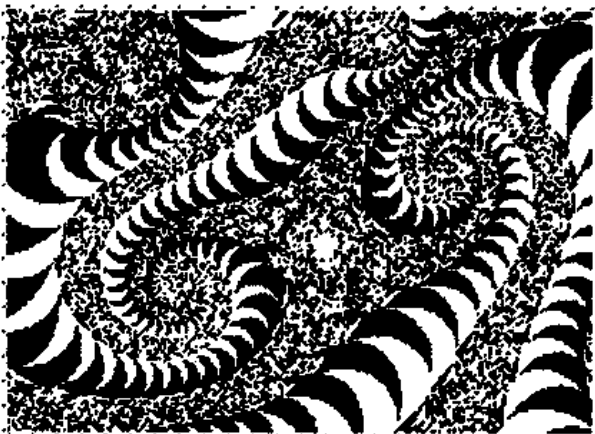
If you are already a Mandelbrot set fan, you will be familiar with the barred structure of the pictures but may notice that the bars are more crescent shaped than you are used to seeing. The probable reason for this discrepancy is that the criterion used by MnJMaps to judge whether the sequence it generates for a given point is not the conventional one. MnJMaps has been designed to compute any Mandelbrot-type map, i.e. any value of $z(0)$ can be selected, and any Julia map, i.e. any value of c can be selected. (My understanding of Julia sets is that the Julia set with parameter c is the set of points $z(0)$ which, for that value of c , yield a bounded sequence). It can be shown that in all these cases the sequence

is unbounded if a value of $|z(n)|$ arises which is greater than $(1 + \sqrt{1 + 4/c})/2$ where $|z(n)|$ and c denote the modulus of $z(n)$ and c respectively. This is the criterion that MnJMaps uses. The above expression evaluates to the conventional criterion value for $c = 2$.



Map 5.

The program consists of some 800 PASCAL text lines. I wrote MnJMaps in PASCAL partly because of the execution speed advantage this approach offers over the corresponding program in BASIC and partly for the experience of learning a high level language and writing a fairly elaborate program in it.



Map 6.

HiSoft's PASCAL package includes a facility for relocating the object code it compiles to addresses formerly occupied by the compiler and for saving to disc the relocated object code and the run-time system as a single file. This file can

subsequently be loaded and the program entered directly from BASIC by means of a RANDOMISE USR command. Relocation saves you the tedium of having to re-compile a program every time you want to run it and releases for possible use by the program a block of RAM (at least 12K bytes) corresponding to that formerly occupied by the compiler, the editor and the PASCAL program text. (In the case of MnJMaps, the latter is a list of 19 PASCAL program text file names which does not occupy much RAM).

I had planned to exploit the block of RAM so released to increase the map file size: this would allow me to reduce the number of files per map from 15 to 3 and to increase the number of maps per disc from 5 to 16. Regrettably there is a bug in the HiSoft package or in the DISCIPLE DOS, or both, which causes the relocated program to crash occasionally while reading a file from disc. HiSoft know about the problem and hopefully will soon find a solution.

Finally, if there is anyone reading this article who would be interested in having a copy of the source files then I am prepared to make them available. You must have the disc version of HiSoft Pascal (available from MGT at around £25), you will also need to be using 3.5" DS/DD 80 track drives as these are the only type I have access to. Send a formatted 3.5" disc together with a stamped addressed envelope to:-

Ray Williams,
77, Heathermount Drive,
Crowthorne,
Berks,
RG11 6HJ.

I should also warn that the program is not very 'user friendly' and you will need to be familiar with Pascal in order to compile and use MnJMaps.

I hope you have enjoyed this short look at Mendelbrot and Julia set pictures. Its a very interesting subject and I have only scraped the surface so far.



YOUR LETTERS



STAR*LETTER* *STAR*LETTER

Dear Editor,

Why do you insist on using DISC instead of DISK in your magazine? All the other magazines spell it the correct way - DISK - but since I joined INDUG just before Christmas I have never seen DISK used in FORMAT.

Yours Sincerely, H.R.Morrison.

Obviously you never had the introductory issue of FORMAT because I explained the reason in there. However I have had several letters on this subject so I will state my side again for the records.

DISK is the American spelling, DISC is the British spelling. I, being British born and bred prefer to use DISC. And as I edit FORMAT its only fair that I get to use my spellings. You dont want me to start using COLOR as well do you?

Still, has anyone else got anything to say on the spelling front? Ed.

Dear Editor,

If there are any members with a PLUS 2a (the black case) then my advice is to take it back to your retailer. Purchasing one from Dixons in early December I have had endless problems both with hardware and software. Taking it back to Dixons I was promptly presented with a PLUS 2 (gray case). The manager would not be drawn into making any comments other than to confirm there were no more PLUS 2As on the shelves.

Yours Sincerely, Allan Stephenson.

Dear Editor,

I always look forward to seeing Carol Brooksbank's articles which I find very interesting and practical. However, when I was prompted to use her 'Small is Beautiful' program (Dec '88) I found the machine code part of the program would not work with the

DISCiPLE / PLUS D.

The use of the call to the ROM printing routine (RST 16) leads to a crash. If this is changed to use the DOS Command Code PNTP (RST 8 followed by DEFB 57). To ammend the machine code delete the first 5 bytes of the code and enter 4 zeros. Then in line 50 of the DATA alter 215 to read 207 and insert 57 immediately after it.

For readers using the STAR LC-10 printer the basic program will also require line 170 changed to LPRINT CHR\$ 10, and the same instruction in a new line 155 to give a line feed.

Yours Sincerely, Ray W.Bray.

Sorry Ray but Carol's program works very well on the PLUS D and DISCiPLE. I know, I use it now whenever I want to print a screen shot for FORMAT. The program allocates the printer channel (#3) and then uses the RST 16 command to print each byte. This method is used because it will work with any Spectrum and any printer interface. I dont know where you went wrong but the program printed in FORMAT is correct.

Your final notes also tell me that you have your printer set up wrong. You need to change the Dip-Switch to give and AUTO [LF] after [CR] (auto line-feed after carrage return). Ed.

Dear Editor,

As a computing on the cheap person, others may be interested in my recent purchase of a 5.25" 780k "Seiko" drive for £39.10 incl p&p and VAT. They are supplied - brand new - from Cambridge Computer Science (Tel 0223 327602) and, although you need to build a power supply, they come with all necessary leads.

Yours Sincerely, Jonathan Butler.

Letters printed may be edited for length or clarity. The writer of each months STAR LETTER wins an EXTRA 6 months subscription to FORMAT.

TOTAL RECALL

By: Bill Long.

Have you ever issued an ERASE d1"*" command by accident? Nasty, all those files just vanish. All that hard work gone forever... Not any longer, RECALL will get them back for you and without you needing to do more than press a few keys.

RECALL looks at each directory entry in turn and if it finds an erased file then it will try to resurrect the file for you. You do not need to know what type of file you ERASEd, RECALL works it out for itself. The theory behind RECALL is that each file type has slightly different details stored in the directory. Remember only the first byte of the directory entry is set to zero to indicate an erased file. The program works through testing for each type of file until it finds a match. It then tests to see that the sectors are still linked together and have not been overwritten. If all is OK then the directory entry is given the correct file descriptor and rewritten to the disc. Hey Presto - one recovered file.

Only one file type can't be recovered in this way and that's the SPECIAL type. These files are only available to machine code programmers and can be anything the programmer wants them to be. If a SPECIAL file is valid by the method used by RECALL it will appear in the CATALOGUE as a normal file, so be warned.

```

1 REM RECALL V1.1
2 REM FOR DISCIPLE & PLUS D.
3 REM (C)1989 FORMAT/INDUG.
4 REM ALL RIGHTS RESERVED.
5 REM
10 DIM T$(11,10)
20 RESTORE : FOR I=1 TO 11: READ T$(
  I): NEXT I
30 DATA "BASIC", "D.ARRAY", "$.ARRAY",

```

```

"CODE", "SNP 48K", "M/DRIVE", "SCREE
N$", "SPECIAL", "SNP128K", "OPENTYP"
, "EXECUTE"
40 CLS #
50 INPUT "DRIVE NUMBER? "; DRIVE
60 IF DRIVE<1 OR DRIVE>2 THEN GOTO 5
  0
70 FOR P=1 TO 80
80 CLS #
90 LET TRACK=INT ((P-1)/20)
100 LET SECTOR=INT ((P/2)-(TRACK*10)+
  0.5)
110 LOAD @DRIVE, TRACK, SECTOR, 50000
120 LET PNT=50000+256*(P/2=INT (P/2))
130 IF PEEK PNT THEN LET P=P+1: GOTO
  90: REM if file not erased then 1
  oop back for next entry
140 IF NOT PEEK (PNT+1) THEN CLS : PR
  INT INVERSE 1; "NO MORE FILES TO
  RECOVER": PAUSE 100: CAT 1: STOP
150 DIM N$(10)
160 FOR I=1 TO 10: LET N$(I)=CHR$ PEE
  K (PNT+I): NEXT I: REM get filena
  me into N$
170 LET SECTORS USED=256*PEEK (PNT+11
  )+PEEK (PNT+12)
180 LET FIRST TRACK=PEEK (PNT+13)
190 LET FIRST SECTOR=PEEK (PNT+14)
200 LET FILE TYPE=PEEK (PNT+211)
210 LET FILE LENGTH=PEEK (PNT+212)+25
  6*PEEK (PNT+213)+65536*PEEK (PNT+
  210)
220 LET START ADDRESS=PEEK (PNT+214)+
  256*PEEK (PNT+215)
230 LET FILE DESCRIPTOR=0
240 IF FILE TYPE=0 AND START ADDRESS<
  >0 THEN LET FILE DESCRIPTOR=1: RE
  M its a basic program.
250 IF FILE TYPE=0 AND SECTORS USED=9
  7 THEN LET FILE DESCRIPTOR=5: REM
  its a 48K snapshot file.
260 IF FILE TYPE=0 AND SECTORS USED=2
  58 THEN LET FILE DESCRIPTOR=9: RE
  M its a 128K snapshot file.
270 IF FILE TYPE=0 AND START ADDRESS=
  0 AND FILE LENGTH<>0 THEN LET FIL
  E DESCRIPTOR=10: REM found an OPE
  NTYPE file.

```

```

280 IF FILE TYPE=3 THEN LET FILE DESC
    RIPTOR=4: REM a CODE file
290 IF FILE TYPE=3 AND START ADDRESS=
    16384 AND FILE LENGTH=6912 THEN L
    ET FILE DESCRIPTOR=7: REM its a S
    CREEN$ file.
300 IF FILE TYPE=2 THEN LET FILE DESC
    RIPTOR=3: REM a string array.
310 IF FILE TYPE=1 THEN LET FILE DESC
    RIPTOR=2: REM a numeric array.
320 IF FILE DESCRIPTOR=0 AND SECTORS
    USED=1 THEN LET FILE DESCRIPTOR=1
    1: REM its an execute file.
330 IF FILE DESCRIPTOR=0 AND SECTORS
    USED>=2 THEN LET FILE DESCRIPTOR=
    6: REM it must be a microdrive ty
    pe file.
340 CLS #: PRINT PAPER 6; INK 0; BRI
    GHT 1;"RECALL v1.1 (c)1988 FORMAT
    /INDUG"
350 PAPER 1: INK 7: PRINT "FILENAME:
    ", PAPER 2;N$,
360 PRINT "'FILE TYPE:", PAPER 2;T$(
    FILE DESCRIPTOR),
370 PRINT "'CATALOGUE No:", PAPER 2;
    P,
380 PRINT "'FILE LENGTH:", PAPER 2;F
    ILE LENGTH,
390 PRINT "'START ADDRESS:", PAPER 2
    ;START ADDRESS,
400 PRINT "'SECTORS USED:", PAPER 2;
    SECTORS USED,
410 PAPER 7: INK 0

```

```

420 PRINT #0;"RECOVER THIS FILE? Y or
    N"
430 POKE 23658,8
440 LET I$=INKEY$: IF I$="" THEN GOTO
    440
450 IF I$="N" THEN GOTO 480
460 IF I$<>"Y" THEN GOTO 440
470 INPUT ,: GOSUB 490
480 NEXT P
490 PRINT #0; FLASH 1;"TESTING FILE S
    ECTORS"
500 LET NEXT TRACK=FIRST TRACK: LET N
    EXT SECTOR=FIRST SECTOR
510 FOR I=1 TO SECTORS USED+1
520 LOAD @DRIVE,NEXT TRACK,NEXT SECTO
    R,50512
530 LET NEXT TRACK=PEEK (50512+510):
    LET NEXT SECTOR=PEEK (50512+511)
540 IF NEXT SECTOR=0 AND NEXT TRACK=0
    THEN GOTO 570
550 NEXT I
560 PRINT AT 20,0; INVERSE 1;"EXTRA S
    ECTORS FOUND": GOTO 610
570 IF I<>SECTORS USED THEN PRINT AT
    20,0; FLASH 1;"NOT ENOUGH SECTORS
    FOUND...": GOTO 610
580 INPUT ,: PRINT #0;"FILE O.K. NO
    W RECOVERING FILE."
590 POKE PNT,FILE DESCRIPTOR
600 SAVE @DRIVE,TRACK,SECTOR,50000
610 PAUSE 100
620 RETURN
9999 SAVE d1"RECALL" LINE 1

```

Bradway Software

Lin-O-Type

Add style to your written word; print out any ASCII wordprocessor file in high density NLQ in a choice of 25 fonts. Turn your Spectrum into an electronic typewriter, superb for addressing envelopes, filling in forms or writing short notes.

- * Full on screen WYSIWYG line editor.
- * Library of 25 fonts including foreign alphabets.
- * Up to 6 fonts may be mixed on each line.
- * Fast, compiled editor to modify & create new fonts.
- * Single or double width characters and underlining.
- * True proportional spacing.
- * Load files from any storage device; mdv, disc, Wafa.
- * Print text as found or edit en route to printer.
- * Double density print gives NLQ in most fonts.
- * Printing is multi-tasked with keyboard use.
- * Optional hexadecimal dump from code files.
- * Price £9.00 (cass) £10.00 (mdv, disc).

Lin-O-Type requires an Epson compatible printer. All Bradway Software programs drive almost any printer interface (including Disciple & Plus D) and are available on 5.25" or 3.5" disc for Discovery or Disciple. Post & Packing: UK & Europe included, please add £1.50 per program world-wide airmail. Payment by cheque, PO, GIRO 65 675 0901, ACCESS. Send for our full catalogue of utility programs for the Spectrum.

WordFinder

At last, help for all you crossword and word game enthusiasts! WordFinder gives you on line access to a large vocabulary to aid those jaded memory cells.

- * 19,000 unique words.
- * 5,000 proper names.
- * Word lengths of 3-12 characters.
- * Search by word skeleton e.g. -g-y.
- * Simple retrieval of plurals, adjectives & adverbs.
- * Superb anagrammer; input more letters than target
- * -or less letters than target (for Scrabble cheats!)
- * Uses Spectrum 128 RAM disc for superfast access.
- * Price £9.50 (cass) £10.50 (disc).

"Hillsett", Upper Padley, Grindleford, Sheffield, S30 1JA. phone (0433) 30799.

HACK-ZONE

By: Hugh J. McLenaghan.

Welcome to another month of Hack-Zone. This month I start with a little more on BASIC protection.

Firstly a little thing to put at the beginning of each line that you do not wish listed. EDIT the line you wish to protect and then press the following keys:-

```
True Video,True Video,True Video
<CAPS SHIFT-5>,<CAPS SHIFT-5>
True Video
<CAPS SHIFT-8>
True Video
```

After doing this the line will vanish. The keys inside the angled brackets have to be pressed at the same time.

Now for a line which will either cause the computer to crash or call a m/c routine if you press BREAK or an error occurs. Make the line one of the first lines the computer executes after the program runs.

```
LET TOP=PEEK 23730+256*PEEK 23731-3:
LET A1=INT(ADDRESS/256):
LET A2=ADDRESS-256*A1:
POKE TOP,A2: POKE TOP+1,A1
```

If ADDRESS points to a machine code routine then a jump will be made to it when an error occurs in Basic. If however, you use zero as the address then the computer will reset itself when the error occurs.

Ever sat down to debug a Basic program only to get well and truly lost with all the GOSUBs and RETURNS? Press BREAK while the program is in the middle of a subroutine and how can you find out how the program got there. What line contains the GOSUB that called the subroutine? Well there

is a way to find out. Just enter the following line at some convenient point in your program.

```
9990 LET PNT=PEEK 23613+256*PEEK 23614
+5:PRINT "Line=";PEEK PNT+256*PEEK (PNT+1),"Statement=";PEEK (pnt+2)-1: STOP
```

Now, whenever you want to know where the GOSUB was, just enter GOSUB 9990 and the line and statement are printed.

Thats all I have for you this month. But before I go I would like to take up some space with an URGENT appeal. I am a university student, often under pressure to produce course-work or swot for exams. As such my spare time is limited. I now have one or two major conversions underway for the HACK ZONE, but these take time to do. In the meantime I'm running out of ideas for things to put in the monthly column. I cant be the only one hacking into programs, there must be hundreds of like minded people reading this page each month. So now its your turn.

I need your tips, ideas, conversions, comments and problems. Without them HACK ZONE may become an irregular page, appearing only when I finish something major. So come on, help me keep HACK ZONE going the way you like it (and if you dont like the things that appear then send in something you do like). Send things to me direct at the following address. If you wish a reply, then include a S.A.E.

Hugh J. McLenaghan (HACK ZONE),
36 Floorsburn Crescent,
Johnstone,
Renfrewshire,
PA5 8PF.

See you next month!

LABEL PRINTER

By: Malcolm Perry.

The following program was written to print out strips of sticky labels. The useful point is that any line can be edited without having to do the whole thing again, to aid this each line is numbered (line numbers not printed). The program as it stands was designed round a star LC10 printer in Pica+Courier mode. This allows for 7 lines of 29 characters to be printed on labels measuring 90mm by 35mm. Obviously the program could be altered for other printers. I have avoided fancy programming so that alterations are easy to make.

To use enter each line as prompted, 'ENTER' alone gives a blank line. Give a "Y" input to the prompt "OK?" to do a print out, any other input or 'ENTER' will put it into edit mode (follow screen prompts).

A "Y" input to the repeat prompt will print another copy of the label.

```

5 CLS : LET R=0: POKE 23658,8
10 INPUT "line 1";A$
20 LET X$=A$
30 LET Y=0: GOSUB 1000
40 IF Y=1 THEN GOTO 10
50 PRINT A$
110 INPUT "line 2";B$
120 LET X$=B$
130 LET Y=0: GOSUB 1000
140 IF Y=1 THEN GOTO 110
150 PRINT B$
210 INPUT "line 3";C$
220 LET X$=C$
230 LET Y=0: GOSUB 1000
240 IF Y=1 THEN GOTO 210
250 PRINT C$
310 INPUT "line 4";D$
320 LET X$=D$
330 LET Y=0: GOSUB 1000
340 IF Y=1 THEN GOTO 310
350 PRINT D$
410 INPUT "line 5";E$

```

```

420 LET X$=E$
430 LET Y=0: GOSUB 1000
440 IF Y=1 THEN GOTO 410
450 PRINT E$
510 INPUT "line 6";F$
520 LET X$=F$
530 LET Y=0: GOSUB 1000
540 IF Y=1 THEN GOTO 510
550 PRINT F$
610 INPUT "line 7";G$
620 LET X$=G$
630 LET Y=0: GOSUB 1000
640 IF Y=1 THEN GOTO 610
650 PRINT G$
660 CLS : PRINT "1- ";A$;"2- ";B$;"3- ";C$;"4- ";D$;"5- ";E$;"6- ";F$;"7- ";G$:
700 PRINT "THIS OK?"
710 INPUT Z$: IF T$="Y" OR T$="y" THEN GOTO 750
730 GOTO 2000
750 LPRINT A$'B$'C$'D$'E$'F$'G$'
760 PRINT AT 17,0;"Repeat?"
770 INPUT T$: IF T$="y" OR T$="Y" THEN GOTO 750
780 CLS : PRINT "new text?"
790 INPUT T$: IF T$="y" OR T$="Y" THEN RUN
800 POKE 23658,0: STOP
1000 IF LEN X$>29 THEN LET Y=1
1005 IF R=1 THEN GOTO 660
1010 RETURN
2000 LET R=1: INPUT "which line?";W$
2011 IF W$="1" THEN LET A$=" ": GOTO 10
2012 IF W$="2" THEN LET B$=" ": GOTO 110
2013 IF W$="3" THEN LET C$=" ": GOTO 210
2014 IF W$="4" THEN LET D$=" ": GOTO 310
2015 IF W$="5" THEN LET E$=" ": GOTO 410
2016 IF W$="6" THEN LET F$=" ": GOTO 510
2017 IF W$="7" THEN LET G$=" ": GOTO 610
2018 GOTO 660

```

ADVENTURE CORNER

By: Paul Rigby.

Well "fellow adventurers" I must say that I have been totally underwhelmed at the lack of response that this column is producing. Are there any adventurers out there? Or have you all gone treasure hunting down the nearest Colossal Cave? When I began this series of ramblings I mentioned that it was up to you to send in your views, suggestions, likes and dislikes. I asked for your feedback. I even said that the success of this adventure column depends on it. I wasn't kidding either. For an adventure column to flourish there has to be a certain amount of feedback from you, the gentle reader. Otherwise it becomes a cold, one-way service that will, sooner or later, become stale.

I have many wonderful and glorious ideas for the future of this column but they all come to nought without your guidance. For example, I have been thinking of a series on PAW programming. But does anyone out there use it? Do you all use the Quill or the GAC? Have you ever used a utility? If not I could develop a beginners guide. Do you want a regular review service of the latest and greatest adventures? A help service on adventures that you are playing? Features on Computer Role Playing Games? Features on adventure history? Less coverage on any subject? Also, how could this column be improved? It is no use silently complaining that I write a load of rubbish. Write in and tell me so we can both do something about it. Please remember this is not a column where Paul Rigby boosts his ego by spouting about anything that takes his fancy. The Adventure Corner is your chance to have a say on what is written. How many other magazines columns are truly guided by the

readers? Not many, I can assure you. So take advantage of the Corner - write in!!

Right - just give me a minute to climb off this soap-box and I'll be right with you. Okay, this month I must make an apology for not reviewing the latest PAW utilities but I promise that they will appear next month. However, to make up for it I thought that I would take a look at the latest version of the PAW, along with the PLUS D/DISCIPLE version. As you may expect, the latter version enables you to use the PAW from disc - probably the way it was meant to be used all along. The update, available for a nominal fee, arrives on tape which is an ideal way of skirting around the problem of variable disc formats that we all use. So PLUS D/DISCIPLE users with 5 1/4", 3 1/2" and 3" discs may all use the disc version of the PAW. The majority of the program is transferred to disc. The files that are left on tape are the Test Adventure, the 22 character sets and the Verify Adventure/Database feature. This move was obviously done to save disc space although I would have liked to have seen the latter feature on the disc to begin with. The useful feature with discs, especially the higher capacity types, is the ability to copy the PAW onto each disc you are working on. Thus negating any disc swaps. The disc version has an extra menu option. The main features of which are to select the current drive, a maximum of two being catered for, the selection of the tape system for saving and loading and a catalogue feature which lists the file saves on the current disc.

Ever since the introduction of the PAW Gilsoft have been quietly

upgrading it. Many users will probably not notice any great difference with the earlier versions. However, version A16 represents a landmark in the evolution of the PAW. With this version Gilsoft set their stall out to provide a structured and documented system of adding "User Overlays". Gilsoft are actively encouraging third parties to produce small additional programs which can be loaded into the PAW's main program as a User Overlay. They are even providing a list of function calls and a description of the database structure for anyone who is interested in writing such a program. Other changes include subtle changes to the "Parse" command, the addition of a new flag and the introduction of a filename system which is used when playing finished games. Thus players can now assign their saved games specific filenames. In addition to that little lot is a Data Management feature which is supplied in the form of an Overlay. Gilsoft describe it as a Hunk Management system. It will allow you to keep control of the data which is being used by the various User Overlays, a feature which will become more valuable as more User Overlays are written for the PAW.

The most exciting feature of the latest version is the "Multi-Parse" commands. Basically, this new feature, developed in association with G.T.Kellet of Kelsoft, allows PSIs to have an increased amount of artificial intelligence. This will give you the ability to direct commands at PSIs who will simultaneously carry them out. To quote an example from the update leaflet;

SAY TO PSI 1 "WAIT, KILL MONSTER"
SAY TO PSI 2 "KILL MONSTER"
KILL MONSTER

All of these instructions would be executed at the same time. However, the Multi-Parse feature is capable of far more complicated actions which involve the PSI completing a variety of actions. The commands which are given to the PSI take the form of Logical Sentences (LS). A list of LS

commands can be typed in by the player with each LS being divided by a comma. The PAW stores each LS and then hands each LS, in order, to the PSI who promptly executes them. Again to take an example from the update leaflet;

"In order to get out of a cavern you need to be lifted on a platform controlled from another room. This can only be achieved by giving a PSI (who happens to be hanging around) a list of things to do."

So a possible solution would be to type in the following;

"GET ON PLATFORM, SAY TO PSI "GO NORTH, PULL CABLE AND RELEASE IT" THEN GET OFF IT."

This just shows a couple of examples for the Multi-Parse feature. But it highlights the advanced position that the PAW is now in. Once you sit down and compare the latest PAW with the likes of Level 9 and Magnetic Scrolls you begin to realise just how powerful the PAW really is. Combine features such as the "DO ALL" command, the use of "IT", the "Multi-Parse" command and the "Extern" feature along with a half-decent imagination and you have an adventure which has the capability to blow anything else out of the water! But as Al once said (or was it Ronnie?) - "you 'aint seen nothing yet!". Catch Format's EXCLUSIVE reviews of Gilsoft's latest User Overlays - PAW-Phosis, PAW-Tel and PAW-Mega - Next Month!

On the subject of Gilsoft. There were rumours about a Gilsoft produced magazine written around the PAW, called "Forge". The magazine was to include hints, tips, news and so on. However, little has been heard of it. I fear that Gilsoft have decided that, through lack of interest, it was not worth the bother. So, if you have a copy of the PAW or are contemplating buying one, why not drop them a line enquiring about the "Forge"? The sudden surge of interest may produce some results. The "Forge" would provide a vehicle to tap the immense experience which Gilsoft holds, to the

benefit of all I am sure.

Which all begs the question. Why don't other software houses, big and small, issue magazines and fanzines? I am sure that it would be the perfect way to get to know the people behind the software. After all nobody knows their games like the publishing software houses do. Imagine in-depth features on your favourite game, interviews with designers, features on development and so on. It is nothing new after all. Infocom do it. They produce their own newspaper called The Status Line (previously The New York Times - until The New York Times got uppity). Infocom provide news of their latest games, features on how their games are made, competitions, behind the scenes stories, etc. Amstrad have their own "Official" magazine, even adventure related companies, such as table-top RPG companies have their own magazines. Games Workshop have their "White Dwarf" magazine while TSR have their "Dragon" magazine. It would be nice to see Magnetic Scrolls, for example produce a similar magazine. Even the smaller adventure companies such as Eighth Day or Topologika would benefit if they produced a fanzine or newsletter. Costs would be a major concern, of course, but with the right ideas and the right

management it would be interesting to see software houses at least experimenting with the idea. In Gilsoft's case, I wonder if it would be worth publishing one large magazine which would cover the PAW and the Quill - and even mention Gilsoft's other associated concerns - such as their part in the making of the SWAN with Fergus McNeill, for example. Has anybody any thoughts on the matter?

See you again next month.

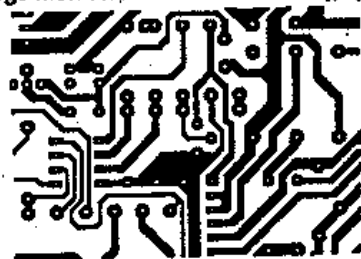
PCB DESIGNER FOR THE 48K ZX SPECTRUM

Now you can produce high quality printed circuit boards/circuit diagrams/component layouts on your 48K ZX Spectrum. If you don't own one it's worth getting one just for this suite of programs! *Comprehensive manual included with getting started tutorial.*

FULL SUITE FOR ONLY £30.00 INC.

PCB LAYOUT:

Produce quality printed circuits directly from your EPSON RX/FX or compatible dot matrix printer using a dense 1:1 printout on positive photoresist coated board. Or super quality using x2 printout and photoreduction. Many features such as 15 track widths; 15 pad sizes; 16 transistor/corners; 20 connectors; large multiscreen WYSIWYG display gives a clear uncluttered view of pads, tracks and drill holes; 0.1in. grid on/off; Block move; copy; mirror; rotate; erase; area fill (ideal for earth plane); preview; undo; dimensionally accurate printer routine with quick print; 1:1 or 2:1 dumps. Custom pad design and library. Available separately for £20.00 inc.

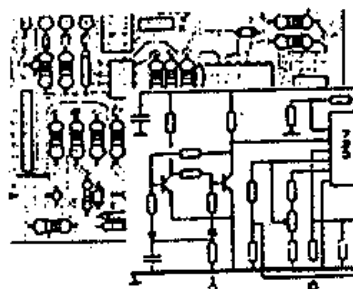


COMPONENT LAYOUT

Draw component layouts directly or from existing pcb layouts using a unique track reducing facility. The following components are provided: resistors, capacitors, ics, diodes, transistors, line drawing, printout and block commands as above. Not available separately.

CIRCUIT DIAGRAMS

Features similar to the above programs with a library of electronic symbols including resistors, capacitors, diodes, transistors, fets, op amp, switches, inductors, logic gates. Not available separately.



State version required from: Disciple/+D; Discovery; +3; Microdrive & Tape. *Important! Tape and Microdrive users please state Centronics interface in use or send £1 for details.*

KEMSOFT THE WOODLANDS, KEMPSEY,
WORCESTER WR5 3NB. Tel. 0905 821088 after 6 p.m., or
see us on A.I.X-386 BULLETIN BOARD 0905 52536/754127
on any computer with modem.



PART 2.

By: Clyde Bish.

In **FORMAT Vol.2 No.7** I started talking about using graphics other than User Defined Graphic (UDGs). Well here is the second part, a month later than promised, after last months foray into **FASTFILE**, but I hope you will consider the wait worth while.

If you want the picture in colour you'll need to add information to the **BOOT** graphics data given in part 1. Again you could use the utility I supplied last time, but I'll give you some to get on with. These are found in **Table C** and will produce a blue and red boot on yellow paper.

TABLE C

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 50 | 50 | 49 | 49 | 49 | 49 | 49 | 49 |
| 49 | 49 | 49 | 49 | 49 | 49 | 49 | 50 |
| 50 | 49 | 49 | 49 | 0 | | | |

Add these extra 20 bytes, one for each character square, with "codeloader", altering the **FOR/TO** values to 60160 and 61079 then **SAVE** the graphics set with:-

SAVE "bootg" CODE 60000,180

You could get these to screen using a **BASIC** program as we have done before, but it would be rather slow. We would do better to resort to some machine code. What we need to do is take each data value, and **POKE** it into the attributes file in blocks of 2 by 2 character squares. Enter the data from **Table D** using "codeloader" with the **FOR/TO** values at 65214 and 65258.

TABLE D

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 33 | 0 | 253 | 17 | 64 | 89 | 14 | 4 |
| 6 | 5 | 197 | 126 | 18 | 19 | 18 | 6 |
| 31 | 19 | 16 | 253 | 18 | 19 | 237 | 160 |
| 6 | 32 | 27 | 16 | 253 | 193 | 16 | 234 |
| 197 | 6 | 54 | 19 | 16 | 253 | 193 | 13 |
| 120 | 177 | 32 | 220 | 201 | 33 | | |

As this is added just before the "newcode" already in you can save it all together with:-

SAVE "BIGCODE" CODE 65214,322

To use this extra code make the following changes to program 3:-

Line 30 - Add **INK 7** to start of the line, and **INK 0** to the end.

Line 40 - Add to the end of the line -
RANDOMIZE s+1312: POKE 65215
,PEEK 23670: POKE 65216,PEEK
23671: RANDOMIZE USR 65214

Now **RUN** and you'll find a multicoloured boot suddenly appears on screen. The reason for the sudden appearance is the **INK 7 ... INK 0**. The boot is drawn in white **INK** on white **PAPER** and so is invisible until the colour is added.

Of course you may not want the boot to appear at line 10, column 0. To move it elsewhere you must change the value **POKE**d into addressed 23306 in line 30 to the number of pixels across (in multiples of 8 - Can you work out why?). The first number in the **POKE 23307** expression is the number of pixels down. You could **PEEK 23689** as suggested earlier and use it to put your illustration on the next available line down.

You will, of course, also have to alter the position of the attribute block. (Now do you see why any pixel moves have to be in multiples of eight!).

Program 4 takes account of these refinements (and also includes the changes mentioned above. Use it to edit Program 3. The changes are:-

Line 25 - A new line which calculates

if there is enough room to print the illustration and scrolls the screen if not.

Line 30 - The value of the variable 1 is first converted to pixels down the screen, then used (in expression 7) to calculate the successive print positions. The number POKEd into address 23306 in the expression which follows is the pixel column for the illustration and can be altered at will. (176 is the limit unless you have a wide screen tele!).

Line 40 - The last four expressions are new. The first calculates the new attribute position using the value in variable 1. The last number in the expression is the PRINT AT column position and must, of course, match the pixel column print position in line 30. For example $80 \div 8 = 10$ here. The remaining expressions set up those values and call the machine code. Anythng which is subsequently printed will appear on the line below the illustration.

PROGRAM 4.

```

1 REM You must CLEAR 65213 & have B
  IGCODE and bootg CODEs on board.
  RUN 9999 to set Code, then use GO
  TO 1
2 LET F$="600000A Boot"
3 GOSUB 20
9 STOP
20 LET S=VAL F$( TO 5)-1152: REM sta
  rt of graphics data less 1152
25 LET L=24-PEEK 23689: POKE 23689,P
  EEK 23689-8: IF L>13 THEN PRINT
  AT 21,0:: FOR F=1 TO L-12: PRINT
  : NEXT F: POKE 23689,3: LET L=13
30 LET L=L*8: INK 7: FOR F=0 TO 3: R
  ANDOMIZE S+40*F: POKE 23606,PEEK
  23670: POKE 23607,PEEK 23671: POK
  E 23307,L+F*16: POKE 23306,0: RAN
  DOMIZE USR 65259: NEXT F: INK 0
40 POKE 23606,0: POKE 23607,60: RAND
  OMIZE S+1312: POKE 65215,PEEK 236
  70: POKE 65216,PEEK 23671: RANDOM
  IZE L*4+22528: POKE 65218,PEEK 23

```

```

670: POKE 65219,PEEK 23671: RANDO
  MIZE USR 65214
60 PRINT F$(6 TO )
70 RETURN
9999 POKE 23308,2: POKE 23309,2: POKE
  23310,8: POKE 23311,144: POKE 233
  12,145: POKE 23313,146: POKE 2331
  4,147: POKE 23315,148: POKE 23316
  ,255

```

You could, of course, use these ideas to produce other shape illustrations. For example a double size 7 by 3 illustration top centre could provide a location scene with text underneath. If you understand the system so far you'll find it quite easy to make the changes.

By now it has probably dawned on a number of readers that the BIGCODE routine (which some pioneer members of the Spectrum fraternity will recognize as a modification of one of those on their "Horizon" tape) could be of use in other situations. In fact anywhere where you want characters printed other than normal size. Program 5 is the subroutine to call when you want to use BIGCODE (which of course you must have on board as well!).

PROGRAM 5.

```

9999 LET A=23306: POKE A,XP: POKE A+1,
  YP: POKE A+2,XM: POKE A+3,YM: POK
  E A+4,8: LET A=A+4: LET L=LEN S$:
  FOR F=1 TO L: POKE A+F,CODE S$(F
  ): NEXT F: POKE A+L+1,255: LET L=
  USR 65259: POKE 23606,0: POKE 236
  07,60: RETURN

```

Before you call it there are certain variables that have to be set. Variable XP is the number of pixels along the horizontal axis where the printing will start, and YP is the number of pixels down the vertical axis. In other words it's like PLOTTing the position of the top left had corner of the start character except that the vertical axis works in reverse. Variable XM is the number of times wider than normal you want the characters, whilst ym is the width-wise magnification. Variable S\$ holds the characters you wish to print. These can be any of the

alphanumeric characters (i.e. those with codes 32 to 127). If you want to use the UDGs (as we did in the boot graphics) remember to alter the Systems Variable called CHARS. If you do this you will need to POKE the value in the ROM (the machines own private memory) for the alphanumeric characters. You will have to redirect this by using POKE 23606, 216 and POKE 23607, 250 before calling the subroutine. After calling you will

* - * - * - *

BACK ISSUES

For members who have missed past issues of FORMAT (or perhaps worn theirs out through constant use) we run a back-issue service. The cost is £1 per issue (£1.25 overseas) incl p&p. Your copies will be sent out with your next monthly issue of FORMAT (provided we receive your order at least a week before). Make cheques (drawn on UK bank or Euro-Cheques, P.O., cash) payable to **FORMAT**.

AVAILABLE ISSUES

Volume 1

Issues #9 (Apr'87) - #12 (Jul'88).

Volume 2

Issues #1 (Aug'88) - #8 (Apr 1989).

Note:- Issues 9 to 12 of Vol 1 are now in short supply.

Please **WRITE YOUR ORDER ON A SEPARATE PIECE OF PAPER**. DO NOT include letters with your order as this will cause delays.

Remember to quote your membership number or we wont be able to send you your order.

BINDERS

COMING SOON - BINDERS FOR YOUR VALUABLE COPIES OF FORMAT. MORE DETAILS NEXT MONTH.

need to POKE the values back to 0 and 60 respectively or any subsequent printing will be unreadable! One last trick with BIGCODE. If you include 9998 LET XP=(256-8*XM*LEN S\$) as a means of setting the value of XP (after of course setting S\$) you will find the printing will be centred on screen.

Well that's all for this issue. Tune in next month for the Big Time!

* - * - * - *

WRITING FOR FORMAT

We are always on the look-out for articles and programs to publish in FORMAT. Articles can be on any subject related to the Spectrum or computing in general. From half a page to a long series.

Don't worry too much about spelling and things like that (the Editor can't spell either) we will sort things out. Just put it down as clearly as you can. It is best if you send your article as a word processor file, on disc or tape, but please include a printed copy so we can look at it straight away. Pack any pictures flat or better still include SCREEN\$ files so we can print them out here.

We are urgently looking for writers to produce articles on the following subjects:-

COMMUNICATIONS

THE 128K ROM

NETWORKING

We also require **REVIEWS** of software and hardware, both new and old. If you want to do a review for us please give us a ring first to confirm no-one else is reviewing the same product.

Send your work to our address on page 2 or give us a ring to talk about it.

INTRODUCTION TO

* FORTH *

By: Alan Cocks.

It was early 1986 and I was beginning to find that the limitations of Basic were getting frustrating. It seemed to me that machine code might be the answer. However, there was a sneaking suspicion that if I immersed myself into machine code I might not surface again.

It was then that I discovered the intriguing language called FORTH.

The reports I read about Forth (it was then the flavour of the month with many magazines) said it had power and many advantages, so I bought a tape-based Forth for my 48k Spectrum.

Over the next few months I hope to give you an insight into Forth and, if I can, encourage you to have a go yourself.

When I first started I carefully followed a book which said it was for beginners and all went well until page 50. Here the sudden suggestion appeared that I should now save some routines onto a Forth "screen". There was no mention of an "editor" etc., or of tapes or discs.

My refuge was the Public library. Four more books on Forth yielded three duds and one gem! This book made sense, with no undue fooling around. Before the stack was hardly mentioned it talked about good and bad Forth systems.

Bad Forth systems? This was where my shock came. Forth was not implemented well into cassette systems! I did not want to believe this. I had spent good money on a cassette based system. It took several weeks for me to come to terms with this. Then I enquired and found that the version I had was not

available for disc or microdrive.

Nor, apparently, was another version I came across. It began to feel like an obstacle course.

Then two good things happened. I found a Forth that was suited to my needs as a beginner - it was used in a Secondary school environment. I also upgraded to microdrives and later to the DISCiPLE and discs.

Forth was written with discs in mind and discs are essential for any serious attempt at using Forth. The key is in the routine which controls reading and writing to disc sectors, known in Forth as BLOCK. It isn't any good just to have this talked about in a hand-book, it has actually got to work with discs in your version.

The Forth standards do not cover editing of source text. This was something else I learned the hard way. A good editor makes a big difference. It became obvious that the Forth standards were formulated by experts, were used and appreciated by experts, and beginners hardly got a look in. What is more, the experts found it easy to write routines to do things like accept keyboard data entry UNDER PROGRAM CONTROL which is something all novices do at an early stage in using Basic. Someone writing assembler routines would expect to put some effort here, or use an existing routine. It came as a surprise to me that Forth did not contain such routines.

Then I came to realise that Forth is a bit more than just a language.

Forth itself is written in Forth! Users are expected to write their own

routines to suit their needs, using the small standard nucleus as the starting point. If your commercial version does not contain a few enhancements, then things may be unduly difficult to start with.

New routines are compiled in sequence, on top of the original nucleus. Programs "run" because each routine contains a pointer. This pointer is the memory address of the relevant previous routine. This leads to fast execution of programs.

Data is handled simply, with a "last in, first out" technique, in an area of memory set aside for use as a "stack".

The simplest way for routines to hand each other data is by means of the stack.

This form of approach demands structured programming techniques and strongly encourages independence of each of the functional modules in the program. Highly desirable for programs you want to live near.

My programming skills, such as they were, came from carefree use of Basic. Oh dear! I could easily pick up the scent of good structure and modularity, yet to achieve it for myself did make the brainbox rattle a bit.

Let's take a closer look. The "stack" is arranged to hold a sequence of numbers. A simple way to put a number onto the stack is to key the number, followed by <ENTER>. For example the sequence:-

7 <ENTER> 3 <ENTER> 42 <ENTER>

will place the three numbers on the stack, with 42 on top, since it was last in.

A number is removed from the stack as it is used, its place being taken by the numbers below it which all move up.

For example, to display the topmost

number, the full stop character is used in Forth. The sequence:-

. <ENTER>

will cause 42 to be displayed on the screen, and the number 42 will be removed from the stack.

Taking the example one stage further, the sequence:-

+ . <ENTER>

will now cause 10 to be displayed on the screen. It is essential to put a space between the characters '+' and '.' since the space is used as a separator character.

The stack is now empty since the topmost number and the next lowest number have been added together, placed at the top of the stack, and from there displayed on the screen.

The location of the stack in memory may easily be discovered in Forth. Although I will not go into it now in detail, it is typical of Forth to allow visibility and also access to its "internal organs".

Taking a close look at the example above, you will see that before the two numbers 3 and 7 were added together, they were both already available and on the stack.

The effective sequence to add them was 7 3 + This is known as "post fix" or "Reverse Polish Notation". It comes only as a result of the stack- and the stack has the benefit of being an attractively simple and economical memory structure.

If you wish, the example can be re-run using each of the other arithmetical symbols in turn; - * / you may notice something a little odd when you come to '/'. This causes the second number on the stack to be divided by the top number, leaving the result at the top of the stack. The result is returned as a whole number, rounded down.

More next month.

SMALL ADS. SMALL ADS.

OVER 1300 Multiface Compatible Pokes on 25+ A4 sheets. Send only £1.30 and 2x1st class stamps to Neil Kurz, 8 Mayfield Close, Old Harlow, Essex, CM17 0LH.

SERIOUS SPECTRUM OWNERS wanting UTILITIES PROGRAMMING HELP, PRACTICAL software and USEFUL articles, GRAPHICS, INFO, IDEAS and MORE from LIKE-MINDED THINKERS, TRY...

OUTLET monthly on MDRIVE, OPUS, DISCIPLE, PLUS D, TAPE

£2 gets YOUR FIRST issue! A blank disc or cartridge (not cassette) gets a FREE demo!

CHEZRON SOFTWARE, 605 LOUGHBOROUGH ROAD, BIRSTALL, LEICESTER LE4 4NJ

PLUS D HACKER Advanced On-Line hacking tool for the PLUS D only. Works with 48k or 128k Spectrums. Includes extended basic commands and much more. As reviewed in Your Sinclair April '89. Send cheque for £4 (£5 overseas) to Steve Nutting, 7, Narrow Close, Histon, Cambridge, CB4 4XX.

DISC CONVERSION SERVICE. Got a game that wont snapshot? Send stamped addressed envelope for list of converted games. By Hugh (Hack Zone) McLenaghan, 36 Floorsburn Crescent, Johnstone, Renfrewshire, PA5 8PF.

BARCELONA. I would like to meet other Spectrum users in my area of Spain. Write to Juan B. Guillen Serra, c/. Mayor de Sarria, 123, 08017 Barcelona, Spain.

TAPESNAP will transfer snapshot files to tape and allow you to reload even without your disc system. Available in two versions TAPESNAP 48 for 48k snaps and TAPESNAP 128 for 128k snaps.

The programs are supplied on tape for auto-save to disc. Price - £4 each or £6 for both incl UK recorded delivery (Overseas +£1). Please send postal orders or cheques (payable to S.Young) to Shimon Young, 21 Colchester Road, Southend-on-Sea, Essex, SS2 6HW.

NOTE: Until late July orders may be delayed due to imminent exams.

Can you

Write programs in Z80 Machine Code ?

Have you

Written any good Spectrum Programs/Utilities in Code or Basic ?

THEN CONTACT DAVE HOOD AT

BETTERBYTES

YOUR ADVERT

Buying, Selling, Pen Friends, etc.

Any PRIVATE advert, up to 30 words (subject to acceptance), will be printed FREE in the next available issue. Any software sold must be original copies, in working order and with full instructions. The publishers will not be held, in any way, responsible for adverts in this column. Trade advertisers should contact the publisher for rates.

NAMES & NUMBERS ALL YOUR DISCS

THE ORIGINAL

Disciple
DISC MANAGER

DISC MANAGER

FOR THE DISCIPLE & PLUS D

NEW
UPGRADE
VERSION

A MUST
FOR ALL
DISCIPLE
& PLUS D
USERS

Plus D
DISC MANAGER
© BETTER BYTES SOFTWARE 1988

INCLUDES THE BEST EVER
2 SEARCH
WAY
IN M/CODE FOR SPECTRUMS

GET TOTAL CONTROL OF YOUR DISC COLLECTION

CUSTOMERS THROUGHOUT
THE WORLD - HERE'S WHAT
SOME OF THEM SAY...

'IMPRESSIVE'

A.S., NEW ZEALAND

'MARVELOUS GRAPHICS'

L.A.R., RENFREWSHIRE

'A DELIGHT TO USE'

F.W.J., CHIPPENHAM

'VERY USEFUL'

R.C.M., NETHERLANDS

'EXCELLENT'

R.H., MIDDLESEX

'ESSENTIAL'

R.W., CALIFORNIA

PROBABLY THE LARGEST
PROGRAM EVER WRITTEN FOR
THE SPECTRUM! COMPRISES
12 PROGRAMS
6 MAIN - 6 SUPPLEMENTARY
A total of 36 files supplied on disc
- over 250K of Program Magic!

The NEW upgrade version of the original DISC MANAGER is the most powerful programme ever written for the DISCIPLE/PLUS D.

Designed to take advantage of Disc Drive ownership, the Manager keeps track of all the programs on all your discs, offering unrivalled benefits and features.

- Storage of up to 27,000 records on one Disc, or 79,920 total. Random File Access.
- Name & Number Discs with fast Autonumber and user pre-defined titles features.
- No typing in of Data. Press a key and Discs are automatically added to appropriate catalogue.
- Fastest ever M/Code Search. 2 modes - Search and Load or Search and List all occurrences, then select program to load.

STOP PRESS

NOW AVAILABLE

DISC ORGANISER

Recover Erased Programs
Re-organise Directory - Repair Sectors - Wild Card Copy, Erase or Re-name
Supplied on Cassette ONLY **£5**

- Special program to print the contents of all your discs with fast sort option. Can selectively print by disc type or number.
- List contents of any disc with ease.
- Multiple erase or rename options.
- Plus many other unique features.
- Comes with 16 page manual and demonstration catalogues.
- Operates with 48K or 128K Spectrums.

Send for the DISC MANAGER today... and you'll soon wonder how you ever Managed without it!

NORMAL PRICE £14.95

SPECIAL INDUG PRICE ONLY

£12.95 INC. P&P

OVERSEAS ORDERS PLEASE ADD £1.50

UPGRADES Existing users can upgrade to the NEW Version.
Send old Manager Disc plus £5.00

BETTERBYTES

10 SPITAL TERRACE · GOSFORTH
NEWCASTLE UPON TYNE NE3 1UT · TEL: (091) 285 6185

SUPPLIED ON 3.5 OR 5.25 DISC · WRITTEN SPECIALLY FOR DISCIPLE/PLUS D

PLEASE STATE YOUR DRIVE TYPE, SIZE, ETC, INDUG MEMBERSHIP No. AND IF REQUIRED FOR DISCIPLE OR PLUS D

EASY TO USE

SPECIAL PRINT-OUT PROGRAM · FLEXIBLE · GREAT GRAPHICS

PERSONALISE YOUR DISCS · AUTOMATIC CATALOGUER · FAST SEARCH & LOAD · PULL DOWN MENUS