MEMOPAD

THE OFFICIAL USER MAGAZINE OF M.C.L

FOR MEMOTECH COMPUTER USERS WORLD WIDE

SPECIAL BIRTHDAY DOUBLE ISSUE

CONTENTS

ISSUE ONE & TWO VOLUME FOUR

Cover Price 2.90p PUBLISHED BY ORION SOFTWARE

VIEWPOINT

Dear Sue,

Please find enclosed short story which you may care to publish in Memopad...

There I was bashing away at the keyboard typing in one of the six 'Football Pools Predictor' programs, when in walks my next door neighbour Mr Knowitall, "and what are we wasting our time on now", he said, peering over my shoulder, What's this? A football forecasting program, you must be crazy - they never work". "This one will" I said, "for a start it has been designed by a learned mathematical professor and cleverly translated by a brilliant programmer", (actually an 80 column maniac). "To add to that this computer is renowned for it's artificial intelligence capabilities", he laughed all the way to the door and thankfully departed.

All the same I knew my task was not easy and I spent many weary hours altering all CSR statements to suit 40 columns and trying to decipher the less legible parts of the listing, but finally it was done, then my disc drive chipped in with one of it's sarcastic comments, bad sector - error on A, so here we go again, copying the five good programs on to a new disc and retyping the one that got away - now the big moment (will it work?), hurrah it does, well after I remember to type in ROM 3.

The next part must be a labour of love, because after typing in 130 team names and 1300 sets of data, you still do not know whether you are for or against, home or away and even what division you are in. Putting the fixtures in is child's play, thankfully, and now we can return to the main menu. Press F4 and wonder of wonders at last here comes the forecast, but what's this - no draws in division 1 or division 2, ah! 3 draws in division 3, coupon number 17, 18 and 20 and more in the next four divisions, all are carefully copied onto the coupon and sent off.

Waiting for Saturday to come is nerve wracking and grandstand drags on for ever but at last the final results are coming through, pen poised - here goes!

No draws in division 1 or 2 and know .. I don't believe it three draws in division three 17,18 and 20, we've cracked it, good old MTX; already I am spending the million pounds - a new series 2, a printer (near letter quality),CP/M, a monitor (colour of course), but oh dear what's going on, it didn't forecast that number, or that stone me only three correct forecasts out of ten results - what went wrong?

It seems that there is no allowance made for accidents, players being sent off weather conditions and all the other factors that can effect the end result and what happened t the Memotech artificial intelligence, I hope it does better next time.

But will it, maybe not, if it knows it is likely to be replaced by the series two, ah well never mind these things were sent to try us - press break and have a kit-kat!

Mr F. Harrison
Membership no: D1675

MEMOPAD

Editorial

Hello Readers,

Well, here we are embarking upon our fourth volume of Memopad. Many of our members have been with us form the magazines inception and must have a mountain of dog- eared back copies by now.

To mark the beginning of our fourth year we are launching a special offer which will run up to X-Mas, see PICK-N-MIX on the back page, also we have had a price review on software and many of our own titles are now down in price. I have also decided to produce software and hardware price list on a bi-monthly basis since I do not think you need a reminder of what we sell every month - but we will still keep you informed on new products and special offers, as and when they come up.

The new video chip from MCL sounds very impressive, I have been asked to point out to you that we do not have any further details ie. price and availability, but as soon as we get the information you will be the first to know.

I should not be telling you this but I knew you would want to know, so here goes. A letter came into the office from one of our software writers detailing a flight simulator he has been working on, it sounds excellent so we have asked for a demo and if this is as good as his description it will definitely be among the additions to our Christmas Catalogue.

So, to conclude my editorial for this birthday issue I am sure you would all like to join me in thanking Keith for his continuing help and support, I certainly do not know what I would do without him!

Sue

ORION SOFTWARE
THE NORTHBRIDGE CENTRE
ELM STREET, BURNLEY
LANCS. BB10 1PD
TELEPHONE - 0282 831695

AVAILABLE BY SUBSCRIPTION ONLY

MEMOPAD IS THE COPYRIGHT OF ORION SOFTWARE

ACCESS AND BARCLAYCARD WELCOME

MEMOPAD

ST-TRUTH

No doubt you are all aware of the latest 'star' on the home computing front - the Atari ST. The recent flurry of adverts and reviews showing all the nice graphics cannot have escaped your notice.

However, having just emerged from a long and traumatic programming session with the said computer my honest opinion is - it is just a glorified Amstrad

The screen handling is atrocious - who ever thought up the method employed by the ST deserves a medal in sadistic mind games. The computer doesn't support hardware sprites. Its has no text mode and all screen i/o is in the bit-mapped mode. High resolution, two colour mode is o.k. but when you get down to the 16 colour mode all is not well - next time you go into a store notice how slowly the screen scrolls.

The low-res mode allows sixteen colours from a palette of 512. Unfortunately the way the screen is mapped on four separate video planes makes it a nightmare for the programmer to update the screen and for smooth action the old method of using a ram screen must be employed - back to the Spectrum days

Basic users, especially the novice, will find it an awesome task to create animated displays. Basic does not support sprites and as a Basic programmer you are required to have an intimate knowledge of the operating system to enable a graphic display to be built up. Peeks and Pokes are a must for the Atari Basic user, and in 32 bits at that

```
;Send a graphic block to the screen ..
          10      start = 491520+(4*640)
          20      for i = start to start + 79 step 2
          30      poke i,65535
          40      next i
```

On the other hand, if you are a C programmer the ST offers tremendous potential.

The Memotech is still one of the most versatile machines on the market. If M.C.L manage to get their act together and overcome a couple of technical problems we will soon be able to offer you an exciting upgrade for your MTX. This will be a new video chip interface. With this new board you will have the facility for the following:-

```
  32 multi coloured sprites
   8 sprites on a line
     internal sprite detection
     block moves within the vdp
     horizontal and vertical raster interrupts.
 512 colours
     512 * 424 pixel resolution
   8 modes of operation including 40 & 80
     column text modes.
```

This represents only a fraction of what the video chip can perform and when the hardware is ready we will have a technical specification ready for publication along with a full review of its capabilities ..... the Memotech is far from dead

MEMOPAD

OOPS!

RE - FOOTBALL POOLS PREDICTOR

Dave sends his profound apologies but it would seem he overlooked the following amendments, he assures me that these are the final changes and when completed you should have a working copy of the program. I think you deserve to win the pools after typing that lot in!

POSITION.SOR

Omit from line 1150 onwards

```
200 CSR 70,0: PRINT DATE$: LET Y=0
210 FOR X=1 TO 25
220 DISC EOF £1,280
230 LET Y=Y+1
240 FOR I=1 TO 11
250 DISC INPUT £1,RECORD$(X,I)
260 NEXT I
270 NEXT X
280 DISC CLOSE £1
290 LET N=INT(Y/2): LET NN=MOD(Y,2): FOR X=1 TO N: CSR 0,X+4: PRINT X;".": CSR 5,X+4: PRINT RECORD$(X,1): CSR 22,X+4: PRINT RECORD$(
X,2): CSR 30,X+4: PRINT X+N;".": CSR 35,X+4: PRINT RECORD$(X+N,1): CSR 53,X+4: PRINT RECORD$(X+N,2): NEXT X
295 IF NN=1 THEN CSR 30,X+4: PRINT X*2-1;".": CSR 35,X+4: PRINT RECORD$((X*2-1),1): CSR 53,X+4: PRINT RECORD$((X*2-1),2)
```

VIEWAMND.FIX

```
1250 DISC OPEN £1,LOA$(2),"I"
1270 LET Y=0: CLS : GOSUB 1000: PRINT "Loading team information": CSR 20,9: PRINT "Now at team number ";Y
1280 DISC INPUT £1,DATE$
1290 FOR X=1 TO 25
1300 DISC EOF £1,1350
1310 LET Y=Y+1: CSR 39,9: PRINT Y
1320 FOR I=1 TO 11
1330 DISC INPUT £1,RECORD$(X,I)
1340 NEXT I: NEXT X
1350 DISC CLOSE £1
1360 RETURN
1400 CLS : CSR 10,0: PRINT "Teams updated to ";DATE$;"     ";AA$;" are for ";WEEK$
1410 GOSUB 1600
1420 CSR 10,3: PRINT DIV$: LET N=Y/2: LET NN=MOD(Y,2)
1430 FOR X=1 TO N
1440 CSR 5,X+4: PRINT X;".": CSR 9,X+4: PRINT RECORD$(X,1): CSR 24,X+4: PRINT N+X;".": CSR 28,X+4: PRINT RECORD$(N+X,1)
1450 NEXT X
1455 IF NN=1 THEN CSR 24,X+4: PRINT X*2-1;".": CSR 28,X+4: PRINT RECORD$((X*2-1),1)
1460 RETURN
```

MEMOPAD

CONTACT3 - REPORT

MODEM: TANDATA TM512

Program loaded with no problem but make sure the user uses the Contact3 disk to startup the computer. Contact was attempted with **Prestel/Micronet** and with various **Bulletin Boards** round the country. Two methods were used:

a) Using the **Dialling facility** in the contact3 menu - no problems
b) Using the internal modem programs. This has two sets of commands - the **HAYES** and **V25 bis.** It didn't seem to matter which one I used, they both worked well - in fact, they seemed to be interchangeable.

Downloading pages from **Prestel** was simple once I found out how to do it, but I had to phone Orion to get the gen. Perhaps this should be included in the **Contact3.doc.** To download a page, either direct to disk or to store in **RAM**, you simply press **<ESC>** followed by **<C>** (but NOT TOGETHER), then using either **PUT** or **STORE.**

COMMANDS

1. **STORE** This command stores the displayed page into **RAM.** It asks you to choose a page number (0-9) then stores the information. On completion it returns to the **COMMAND MODE** - simply press **<T>** to continue with session. This is the quickest way to store information during a session - should there be more than 9 pages you simply use **WRITE** after storing page 9, then once file is on disk, use **CLEAR** to clean pages, then press **<T>** and continue.

2. **CLEAR** This command erases **ALL** stored **RAM** pages and replaces the information with blanks.

3. **WRITE** This dumps stored **RAM** pages to disc under **ONE** file name - a prompt asks you for the name - I think it is a good idea to get into the habit of using **CAPITALS** for the file names. The command can be used either at the end of a session or during it if the information needs more than the 10 pages.

4. **READ** This does the opposite of **WRITE.** When pressed, a prompt asks you for the file name, then loads file into **RAM** in same form as it was stored - ie in **RAM** pages. If you get a flashing **%Disc I/O error** message, simply press **<ESC>.** This can happen if you have typed the file name in lower case and it is in upper case or vice-versa.

5. **BROWSE** When you have finished, this command allows you to view the pages you stored during the session. If you have already put some pages on disk then you will have to use **READ** to get them back. It flicks through the pages in a continuous loop until any key is pressed.

6. **LOAD** When there are **RAM** pages in memory, this command allows to you to see a particular page by answering the prompt **Memory (0-9).**

7. **PUT** As an alternative to **STORE** you can dump the page straight to disc by using this command. Press **<ESC>** then **<C>** (but **NOT at same time**). Pressing **<P>** will bring the prompt for a file name then stores the file on disc under that name. It returns to command mode, then press **<T>** and continue. As in **WRITE**, it is a good idea to get into the habit of always using capitals for file names.

8. **GET** Is the opposite of **PUT**. This brings the prompt for the file name then retrieves the page from disc and displays it directly on to the screen. Like **READ** you may get a flashing error message but just press **<ESC>** and try either lower or upper case instead of what you used first time.

9. **DIAL** Simply press **<D>** then when prompted enter number and press **<RET>**.

10. **VIEWDATA MODE** This allows you to edit a page being displayed on the screen, or even to make up a **RAM** page of your own. I couldn't make much of this function - no doubt someone will explain it to me some time.

11. **MODEM MODE** Puts you on line with the modem without going into **TERMINAL MODE** - I think. At least when I was in this mode, I gave the command to dial - it returned to command mode, then rang the number. By pressing **<T>** the contact was made. As to whether it passed other messages on I don't know. It simply returned to command mode but there was no modem reply to be seen anywhere.

12. **TERMINAL MODE** Puts you on line with the modem - you can send it commands and it replies. I was able to use the directory facility with the modem and can dial and log on to **PRESTEL** automatically. All of the commands in the modem handbook seemed to work OK in this mode. This is the mode used to communicate with other computers.

COMMENTS

1. There is no way to clear the screen simply. It is difficult to see the commands you give to the modem and its replies. On the opening title page, the letters sometimes appear as graphics symbols. You can get a clear screen by using **BROWSE** but on my monitor a **PHILLIPS Computer Monitor 80** the writing is yellow and it is difficult to see against the yellow page background.

2. There is no facility for obtaining a hard copy. With files obtained by using **PUT**, it is possible to "type" some of them by using the CP/M facility + CONTR-P, or by using **NEWWORD** but not in every case. There are also a lot of control codes to be edited out. Could there be some form of screen dump built in?

3. All files stored on disc either by **PUT** or **STORE** use the same extension - **.VID**. Unless you have a separate note of which are which, you might press the wrong key when wanting to see them. How about using the extension **.RAM** for files obtained by using **STORE**?

4. As yet there is no facility for downloading or uploading files to and from libraries and bulletin boards. **PDSIG Library** has lots of games and utilities which can be downloaded and other bulletin boards do the same. Hopefully this can also be built in to the system.

5. I experimented with Bulletin Boards in different parts of the country and in most cases had no difficulty in logging on to them. Where they used the **Prestel** format, there was no problem in getting, storing and later retrieving information. Where a continuous transmission was used, ie at foot of page the cursor went back to the top of the screen and continued, the problem was that the previous page was still there and of course it became very difficult to read the new information. Perhaps something could be added to the program to take care of this are there seems to be a lot of information available for the cost of a telephone call.

6. I think another good idea would be to show the cursor eg when you are putting in modem commands or editing pages or even when putting in numbers to the modem's directory.

7. When making a working copy of **CONTACT3**, after formatting the blank disk, then use the **CONTACT3** master disk to put the system tracks on it.

8. While I presume **CONTACT3.DOC** will appear with the disk, perhaps it could contain a bit more explanation for inexperienced users. For example, how to download a page from **PRESTEL**, not to press **<ESC>** and another key at the same time, and what "**<ESC>+<ESC>...Send 1 (ESCAPE) $1B**" does - I didn't dare try this one!

All-in-all, **Contact3** is an easy to use program once you have mastered the basics. It will give the facility of contacting **PRESTEL, MICRONET, TELECOM GOLD and BULLETIN BOARDS** without having to invest in another piece of equipment like the speed splitter. I have had great fun using it and am making arrangements to do my banking and bill-paying through **HOBS** in Prestel. However, I wait with some trepidation, my next telephone bill. I look forward to seeing an amended version containing the points I have noted above. Who knows? Perhaps we might even get a MTX base in Micronet!

We sent the reviewer a development copy to try under a user orientated environment. He came up with some very valid points, which are listed in the above un-edited report. With his comments in mind we have made the necessary amendments and the finished version is now available.

CONTACT III
NOW AVAILABLE

NO ADD ON HARDWARE IS REQUIRED

CONTACT *PRESTEL : MICRONET 800 ETC....*

SUITABLE FOR ANY MODEM WHICH UTILISES 1200/1200 AT FULL DUPLEX .... TANDATA, WS2000 ETC.

HANDLES ALL GRAPHIC CODES EXCEPT DOUBLE HEIGHT AND WILL RECEIVE IN COLOUR OR MONOCHROME. AUTO-DIAL FUNCTION & MANY MORE USEFUL UTILITIES SUCH AS FACILITY TO LOCALLY EDIT SCREEN AND SAVE YOUR FAVOURITE SCREENS TO DISC.

CPM MODELS ONLY. PLEASE STATE CONFIGURATION 14.95p **INCLUSIVE.**

Minuet in 'B' Flat

```
1 REM MENUETE 1: (page 14). J.S. BACH
5 DIM N(67)
6 LET N(1)=1015.0372
7 LET N(2)=N(1)/1.062
8 LET N(3)=N(1)/1.124
9 LET N(4)=N(1)/1.191
10 LET N(5)=N(1)/1.262
11 LET N(6)=N(1)/1.338
12 LET N(7)=N(1)/1.417
13 LET N(8)=N(1)/1.501
14 LET N(9)=N(1)/1.591
15 LET N(10)=N(1)/1.685
16 LET N(11)=N(1)/1.786
17 LET N(12)=N(1)/1.892
18 LET N(13)=N(1)*.5
19 FOR X=13 TO 66
20 LET N(X)=N(X-12)/2
21 NEXT X
22 LET N(67)=8
23 LET H=204
24 LET A=15: LET B=8: LET C=12
25 PRINT : PRINT : PRINT : PRINT
26 PRINT : PRINT
27 PRINT "            MINUET in B flat"
29 PRINT : PRINT
30 PRINT "              J. S. BACH"
31 PAUSE 1000
36 GOSUB 52
38 LET H=204
40 RESTORE 1016
42 FOR X=1 TO 3: GOSUB 990: NEXT X
44 LET H=102
46 FOR X=1 TO 6: GOSUB 990: NEXT X
48 GOSUB 52
50 GOTO 58
52 LET H=204: RESTORE 1001
54 FOR X=1 TO 90: GOSUB 990: NEXT X
56 RETURN
58 LET H=204: RESTORE 1017
60 FOR X=1 TO 6: GOSUB 990: NEXT X
62 GOSUB 78
64 GOTO 66
66 LET H=204: RESTORE 1039
68 FOR X=1 TO 2: GOSUB 990: NEXT X
70 LET H=102
72 FOR X=1 TO 8: GOSUB 990: NEXT X
74 GOSUB 78
76 GOTO 92
78 LET H=204: RESTORE 1018
80 FOR X=1 TO 45: GOSUB 990: NEXT X
82 LET H=102
84 FOR X=1 TO 6: GOSUB 990: NEXT X
86 LET H=204
88 FOR X=1 TO 78: GOSUB 990: NEXT X
90 RETURN
92 LET H=204: RESTORE 1040
94 FOR X=1 TO 6: GOSUB 990: NEXT X
95 PAUSE 1000: CLS
96 SOUND 0,0,0: SOUND 1,0,0: SOUND 2,0,0
98 GOTO 98
988 SOUND 0,0,0: SOUND 1,0,0: SOUND 2,0,0
989 GOTO 989
990 READ V1,V2,V3
992 SOUND 0,N(V1),A: SOUND 1,N(V2),B: SOUND 2,N(V3),C
993 PAUSE H
994 RETURN
1001 DATA  40,67,24,36,67,24,35,67,19,36,67,19,31,67,16,36,67,16
1002 DATA  40,67,12,67,36,12,41,67,14,67,36,14,43,67,16,67,36,16
1003 DATA  38,67,11,35,67,11,33,67,14,35,67,14,31,67,19,35,67,19
1004 DATA  38,67,23,67,31,23,40,67,24,67,31,23,41,67,21,67,31,19
1005 DATA   40,67,24,34,67,24,41,67,26,33,67,26,43,67,28,31,67,28
1006 DATA  45,67,29,43,67,29,47,67,31,41,67,31,48,67,33,40,67,33
1007 DATA  38,67,35,41,67,35,40,67,36,38,67,36,40,67,24,36,67,24
1008 DATA  38,67,31,35,67,31,33,67,19,35,67,19,31,67,23,41,67,23
1009 DATA   40,67,24,36,67,24,35,67,19,36,67,19,31,67,16,36,67,16
1010 DATA   40,67,12,67,36,12,41,67,14,67,36,14,43,67,16,67,36,16
1011 DATA   38,67,11,35,67,11,33,67,14,35,67,14,31,67,19,35,67,19
1012 DATA   38,67,23,67,31,23,40,67,24,67,31,23,41,67,21,67,31,19
1013 DATA  40,67,24,38,67,24,42,67,26,36,67,26,43,67,28,35,67,28
1014 DATA  33,67,18,36,67,18,31,67,16,36,67,16,30,67,14,36,67,14
1015 DATA  35,67,19,38,67,19,35,67,12,31,67,12,26,67,14,30,67,14
1016 DATA  31,67,7,31,67,11,31,67,14,33,67,19,35,67,19,36,67,21,35,67,21,36,67,23,38,67,23
1017 DATA  31,67,19,31,67,19,31,67,14,31,67,14,31,67,11,31,67,11
1018 DATA  47,67,7,43,67,7,42,67,14,43,67,14,38,67,19,43,67,19
1019 DATA  47,67,17,67,43,17,48,67,16,67,43,16,47,67,12,67,43,12
1020 DATA  45,67,17,67,41,17,40,67,21,67,41,21,38,67,22,67,41,22
1021 DATA  43,67,16,67,40,16,43,67,21,67,38,21,43,67,9,67,37,9
1022 DATA  38,67,14,41,67,14,44,67,23,38,67,23,47,67,21,38,67,21
1023 DATA   35,67,20,67,38,20,32,67,16,67,38,16,28,67,20,67,38,20
1024 DATA  36,67,21,41,67,21,40,67,12,36,67,12,38,67,16,35,67,16
1025 DATA  33,67,9,33,67,11,33,67,12,35,67,14,36,67,14,35,67,16,33,67,16,31,67,17,29,67,17
1026 DATA  28,67,19,36,67,19,35,67,17,36,67,17,43,67,16,36,67,16
1027 DATA  46,67,19,67,36,19,45,67,17,67,36,17,43,67,16,67,36,16
1028 DATA   29,67,21,36,67,21,35,67,19,36,67,19,45,67,17,36,67,17
1029 DATA  29,67,21,67,36,21,31,67,19,67,36,19,33,67,17,67,36,17
1030 DATA  38,67,22,36,67,22,40,67,24,34,67,24,41,67,26,33,67,26
1031 DATA  43,67,28,41,67,28,45,67,29,40,67,29,47,67,31,38,67,31
1032 DATA  48,67,33,47,67,33,45,67,35,43,67,35,41,67,36,40,67,36
1033 DATA   38,67,31,36,67,31,35,67,19,33,67,19,31,67,23,29,67,23
1034 DATA   28,67,28,43,67,28,45,67,16,43,67,16,38,67,19,43,67,19
1035 DATA  37,67,21,67,43,21,35,67,23,67,43,23,33,67,25,67,43,25
1036 DATA  26,67,26,41,67,26,43,67,14,41,67,14,36,67,17,41,67,17
1037 DATA  35,67,19,67,41,19,33,67,21,67,41,21,31,67,23,67,41,23
1038 DATA  40,67,24,43,67,24,40,67,17,36,67,17,31,67,19,35,67,19
1039 DATA  36,67,12,36,67,16,36,67,14,38,67,14,40,67,12,41,67,12,43,67,11,42,67,11,43,67,9,45,67,9
1040 DATA  36,67,12,36,67,12,36,67,12,36,67,12,36,67,12,36,67,12
```

DATABASE

```
750 NEXT M
760 NEXT I
770 NEXT K
780 IF A=128 THEN  GOSUB 1350
790 USER OPEN£1,FILE$(3),"O"
800 CLS : CSR 20,2: PRINT "Sorting completed"
810 CSR 20,5: PRINT "Now saving sorted data.....": CSR 20,7: PRINT "Please wait....."
820 CSR 20,10: PRINT "Now saving record number "
830 IF A=129 AND V=0 THEN  GOTO 1400
840 FOR I=1 TO Z: CSR 45,10: PRINT I
850 FOR N=1 TO W
860 USER  PRINT £1,RECORD$(I,N)
870 NEXT N: NEXT I
880 USER CLOSE£1
890 CLS : CSR 25,5: PRINT "Saving completed": PAUSE 1000
900 IF T=1 THEN  GOTO 940
910 USER ERAD$
920 USER RENO$=FILE$(3)
930 GOTO 960
940 USER ERAFILE$(1)
950 USER RENFILE$(1)=FILE$(3)
960 CLS : GOTO 420
1000 CLS : CSR 25,0: PRINT O$: CSR 24,1: PRINT "=========="
1010 CSR 20,5: PRINT "Please wait....."
1020 RETURN
1050 CLS : CSR 20,10: PRINT "Is file a search list? (Y/N)": GOSUB 1200
1060 IF A$="n" OR A$="N" THEN  GOTO 1090
1070 USER OPEN£1,FILE$(1),"I"
1080 LET T=1: GOTO 1100
1090 USER OPEN£1,O$,"I"
1100 RETURN
1150 FOR X=0 TO 79: PRINT "-";: NEXT X: RETURN
1200 LET A$=INKEY$: IF A$<>"" THEN  GOTO 1200
1210 LET A$=INKEY$: IF A$="" THEN  GOTO 1210
1220 IF A$<>"n" AND A$<>"N" AND A$<>"Y" AND A$<>"y" THEN  GOTO 1200
1230 RETURN
1250 LET A$=INKEY$: LET A=ASC(A$): RETURN
1300 CLS : CSR 28,0: PRINT O$: CSR 27,1: PRINT "==========": RETURN
1350 CLS : CSR 25,5: PRINT F$(1);"Low-to-high"
1360 CSR 25,7: PRINT F$(2);"High-to-low"
1370 GOSUB 1250: IF A<128 OR A>129 THEN  GOTO 1370
1380 RETURN
1400 FOR I=Z TO 1 STEP -1
1410 CSR 45,10: PRINT "    ": CSR 45,10: PRINT I
1420 FOR N=1 TO W
1430 USER  PRINT £1,RECORD$(I,N)
1440 NEXT N: NEXT I
1450 GOTO 880
```

MEMOPAD

```
10 REM ##########################################################################
20 REM ############################## PRINT.BAS  ##############################
30 REM ############################## VERSION 2  ##############################
40 REM ############################## D. WEMYSS  ##############################
50 REM ##############################  OCT 1986  ##############################
60 REM ##########################################################################
70 USER  SAVE "PRINT.BAS"
80 CLEAR : VS 5: CLS : CSR 20,5: PRINT "Please wait.....": CSR 20,7: PRINT "Setting up variables"
90 DIM A$(3),FILE$(2,12),C$(2,18),HEAD$(10,8),RECORD$(80,10,25),O$(8),B$(12),DATE$(9),F$(2,14),D$(2,19),O(10),PL$(21),PR$(23),NAME$(9)
100 LET W=0: LET T=0: LET Y=0: LET Z=0: LET DP=0: LET IT=0: LET E=0: LET LB=0: LET B$="": LET PL$="Please wait.....": LET PR$="Printing record number "
110 CLS : CSR 13,5: INPUT "Which file to print? )";B$: IF LEN (B$))8 THEN  GOTO 110 ELSE  LET O$=B$
120 LET FILE$(1)=O$+".SER": LET FILE$(2)=O$+".HDS"
130 GOSUB 1850
140 CSR 25,7: PRINT "Loading heading number ";W
150 USER OPEN£1,FILE$(2),"I"
160 FOR I=1 TO 10
170 USER EOF£1,210
180 LET W=W+1: CSR 48,7: PRINT W: PAUSE 100
190 USER  INPUT £1,HEAD$(I)
200 NEXT I
210 USER CLOSE£1
220 GOSUB 1900
230 GOSUB 1850
240 CSR 25,7: PRINT "Loading record number ";Y
250 FOR I=1 TO 200
260 USER EOF£1,310
270 LET Y=Y+1: CSR 47,7: PRINT Y: PAUSE 100
280 FOR N=1 TO W
290 USER  INPUT £1,RECORD$(I,N)
300 NEXT N: NEXT I
310 LET Z=Y
320 USER CLOSE£1
330 LET F$(1)="F1.....": LET F$(2)="F2.....": LET D$(1)="Print data": LET D$(2)="Return to Main Menu": LET C$(1)="Print Data Routine": LET C$(2)
=O$+" by D.W."
340 CLS : CSR 28,0: PRINT C$(1): CSR 28,2: PRINT C$(2): PRINT : GOSUB 2500
350 LET N=0
360 FOR I=7 TO 10 STEP 3: LET N=N+1
370 CSR 25,I: PRINT F$(N);D$(N)
380 NEXT I: PRINT : GOSUB 2500
390 LET A$=INKEY$: LET A=ASC(A$): IF A<128 OR A>129 THEN  GOTO 390
400 IF A=128 THEN  GOTO 500 ELSE  GOSUB 3100
410 CLS : CSR 20,10: PRINT "Returning to Main Menu": CSR 20,12: PRINT PL$: LPRINT CHR$(27);"@"
420 USER  LOAD "MAINMENU.BAS"
500 REM Print File Routine
505 GOSUB 2200: LPRINT CHR$(27);"R";CHR$(3)
510 PLOD "PROG1"
520 GOSUB 2000: LET A=ABS(A-128)
530 ON A GOTO 550,700,900,1100,1200,1500,340
550 REM Print List of Records
560 GOSUB 2850: GOSUB 2100
570 GOSUB 2350: GOSUB 3400
580 CSR 0,5: PRINT CHR$(5): CSR 20,5: PRINT PL$
590 LPRINT CHR$(27);"D";CHR$(1);CHR$(6);CHR$(32);CHR$(58);CHR$(0);CHR$(27);"M";CHR$(1)
600 FOR Q=1 TO C: GOSUB 2700
610 CSR 20,8: PRINT "Now printing listing "
620 FOR I=1 TO Z: CSR 41,8: PRINT I
630 LPRINT I,RECORD$(I,1),RECORD$(I,2),RECORD$(I,3)
640 NEXT I: NEXT Q
650 LPRINT CHR$(27);"D";CHR$(0)
```

```
660 GOTO 2750
700 REM Print Individual Record
710 GOSUB 2850: GOSUB 2100
720 CLS : GOSUB 2350: GOSUB 3400
725 GOSUB 3600
730 CLS : CSR 13,5: INPUT "Which record do you wish to print? ) ";I: IF I<1 OR I>Z THEN  GOTO 730
740 FOR X=1 TO W
750 CSR 20,X+7: PRINT RECORD$(I,X): NEXT X
760 PRINT : GOSUB 2500
770 CSR 25,20: PRINT "Is this the correct record? (Y/N)": GOSUB 2050
780 IF A$="N" OR A$="n" THEN  CLS : GOTO 720
790 CLS : GOSUB 2350
800 CSR 25,7: PRINT PL$: CSR 25,9: PRINT PR$;I
810 GOSUB 2700
820 FOR Q=1 TO C: FOR X=1 TO W
830 LPRINT RECORD$(I,X)
840 NEXT X: FOR X=1 TO 3: LPRINT : NEXT X: NEXT Q
850 CLS : CSR 5,20: PRINT F$(1);"Print another record       ";F$(2);"Return to menu"
860 LET A$=INKEY$: LET A=ASC(A$): IF A<128 OR A>129 THEN  GOTO 860
870 IF A=129 THEN  GOTO 2750
880 GOTO 725
900 REM Print all Records
910 GOSUB 2850: GOSUB 2100: GOSUB 2350: GOSUB 3400
920 LPRINT CHR$(27);"D";CHR$(1);CHR$(5);CHR$(35);CHR$(40);CHR$(0);CHR$(27);"N";CHR$(1)
930 FOR Q=1 TO C: GOSUB 2700
940 LET K=1
950 CLS : CSR 25,5: PRINT PL$: CSR 25,7: PRINT "Printing records"
960 CSR 42,7: PRINT K;" and ";K+1
970 LPRINT K,RECORD$(K,1),K+1,RECORD$(K+1,1)
990 FOR X=2 TO W: LPRINT "   ",RECORD$(K,X),"   ",RECORD$(K+1,X)
1000 NEXT X
1010 LET K=K+2
1020 LPRINT : LPRINT : IF K<Z THEN  GOTO 960
1030 IF K>Z THEN  GOTO 1080
1040 CSR 0,7: PRINT CHR$(5): CSR 25,7: PRINT PR$;K
1050 LPRINT K,RECORD$(K,1)
1060 FOR X=2 TO W: LPRINT "   ",RECORD$(K,X)
1070 NEXT X
1080 NEXT Q: LPRINT CHR$(27);"D";CHR$(0)
1090 GOTO 2750
1100 REM Print Record Format
1110 GOSUB 2850: GOSUB 2100
1120 CLS : CSR 25,5: PRINT PL$: CSR 25,7: PRINT "Printing record format"
1130 LPRINT "Format of each record": GOSUB 2550: LPRINT
1140 FOR X=1 TO W
1150 LPRINT HEAD$(X)
1160 NEXT X: GOTO 2750
1200 REM Print Search List
1210 GOSUB 3550: CLS : GOSUB 2850: GOSUB 2100
1220 CLS : GOSUB 2350: GOSUB 3400
1230 PLOD "PROG2"
1240 GOSUB 2000: LET A=ABS(A-128)
1250 ON A GOTO 1260,1350,1390,1430,1480
1260 LET H=1: GOSUB 2250
1270 FOR Q=1 TO C: GOSUB 2700: CLS : CSR 20,5: PRINT FILE$(1);"  Column  ";HEAD$(L)
1280 CSR 20,10: PRINT PL$: CSR 20,12: PRINT PR$
1290 FOR I=1 TO Z: CSR 43,12: PRINT I
1300 CSR 0,14: PRINT CHR$(5): CSR 20,14: PRINT RECORD$(I,L): LPRINT RECORD$(I,L)
1310 NEXT I: NEXT Q: GOTO 2750
1350 LET H=2: GOSUB 2250
```

MEMOPAD

```
1360 FOR Q=1 TO C: GOSUB 2700: CLS : LPRINT CHR$(27);"D";CHR$(1);CHR$(30);CHR$(0)
1370 GOSUB 2350: GOSUB 2400
1380 NEXT Q: LPRINT CHR$(27);"D";CHR$(0): GOTO 2750
1390 LET H=3: GOSUB 2250
1400 FOR Q=1 TO C: GOSUB 2700: CLS : LPRINT CHR$(27);"D";CHR$(1);CHR$(30);CHR$(60);CHR$(0)
1410 GOSUB 2350: GOSUB 2450: GOSUB 2400
1420 NEXT Q: LPRINT CHR$(27);"D";CHR$(0);CHR$(18): GOTO 2750
1430 LET H=4: GOSUB 2250
1440 FOR Q=1 TO C: GOSUB 2700: CLS : LPRINT CHR$(27);"D";CHR$(1);CHR$(30);CHR$(60);CHR$(90);CHR$(0)
1450 GOSUB 2350: GOSUB 2450: GOSUB 2400
1460 NEXT Q: LPRINT CHR$(27);"D";CHR$(0);CHR$(18)
1470 GOTO 2750
1480 GOSUB 3450
1490 GOTO 2750
1500 REM Print Labels
1510 GOSUB 2850: GOSUB 2350
1520 CLS : CSR 20,5: INPUT "Number of lines to print on label? ) ";H: IF H<1 OR H>W THEN  GOTO 1520
1530 PRINT : GOSUB 2500: LET LB=1
1540 GOSUB 2250: CLS : GOSUB 2350
1550 IF H=1 THEN  CSR 25,5: PRINT "Column to print is ";HEAD$(L)
1560 PAUSE 2000: GOTO 1600
1570 CSR 25,5: PRINT "Columns to print are "
1580 FOR I=1 TO H: CSR 25,I+6: PRINT HEAD$(O(I))
1590 NEXT I: PAUSE 2000
1600 LPRINT CHR$(27);"D";CHR$(1);CHR$(5);CHR$(45);CHR$(0)
1610 GOSUB 2100: CLS : GOSUB 2350
1620 CSR 25,5: PRINT PL$: CSR 25,7: PRINT PR$: LET K=1
1630 GOSUB 2600
1640 CSR 48,7: PRINT K;" and ";K+1
1650 FOR I=1 TO H
1660 LPRINT " ";RECORD$(K,O(I)),RECORD$(K+1,O(I))
1670 NEXT I: LPRINT : LET K=K+2
1680 IF H=10 OR H=9 THEN  LPRINT
1690 IF H=8 OR H=7 THEN  LPRINT : LPRINT
1700 IF H=6 OR H=5 THEN  FOR X=1 TO 3: LPRINT : NEXT
1710 IF H=4 OR H=3 THEN  FOR X=1 TO 4: LPRINT : NEXT
1720 IF H=2 OR H=1 THEN  FOR X=1 TO 5: LPRINT : NEXT
1730 IF K<Z THEN  GOTO 1630
1740 IF K>Z THEN  GOTO 1800
1750 CSR 48,7: PRINT K;"
1760 GOSUB 2600
1770 FOR I=1 TO H
1780 LPRINT RECORD$(K,O(I))
1790 NEXT I
1800 LPRINT CHR$(27);"D";CHR$(0)
1810 GOTO 2750
1850 CLS : CSR 32,0: PRINT O$: CSR 31,1: PRINT "=========="
1860 CSR 25,5: PRINT PL$
1870 RETURN
1900 CLS : CSR 20,10: PRINT "Is file a search list? (Y/N)": GOSUB 2050
1910 IF A$="N" OR A$="n" THEN  GOTO 1940
1920 USER OPEN£1,FILE$(1),"I"
1930 LET T=1: GOTO 1950
1940 USER OPEN£1,O$,"I"
1950 RETURN
2000 LET A$=INKEY$: IF A$<>"" THEN  GOTO 2000
2010 LET A$=INKEY$: IF A$="" THEN  GOTO 2010
2020 LET A=ASC(A$): IF A>90 AND A<127 THEN  LET A=A+32: LET A$=CHR$(A)
2030 RETURN
2050 LET A$=INKEY$: IF A$<>"" THEN  GOTO 2050
2060 LET A$=INKEY$: IF A$="" THEN  GOTO 2060
```

MEMOPAD

```
2070 IF A$()"N" AND A$()"n" AND A$()"Y" AND A$()"y" THEN  GOTO 2050
2080 RETURN
2100 CLS : CSR 25,5: PRINT "PLEASE CHECK.....": PAUSE 1000
2110 CSR 25,7: PRINT "1.  Printer is on line": PAUSE 1000
2120 IF LB=0 THEN  GOTO 2150
2130 CSR 25,9: PRINT "2.  Labels are loaded": PAUSE 1000
2140 CSR 25,11: PRINT "3.  1st Label is lined up correctly"
2150 CSR 25,20: PRINT "Ready to start? (Y/N)": GOSUB 2050
2160 IF A$="N" OR A$="n" THEN  GOTO 2100
2170 RETURN
2200 CLS : CSR 25,5: PRINT "Enter date of printout"
2210 CSR 25,20: PRINT "Use format 03 OCT 86": PAUSE 2000
2220 LET B$="": CSR 25,10: INPUT "> ";B$: IF LEN (B$))9 THEN  GOTO 2220 ELSE  LET DATE$=B$
2230 CLS : RETURN
2250 CLS : GOSUB 2350: FOR I=1 TO W: CSR 25,I+2: PRINT I: CSR 30,I+2: PRINT HEAD$(I): NEXT
2260 CSR 25,20
2270 IF H=1 THEN  INPUT "Which column to print? > ";L: GOTO 2300
2280 IF H)1 THEN  PRINT "Which columns to print?"
2290 FOR X=1 TO H: CSR 25,X+13: INPUT "> ";O(X): NEXT
2300 RETURN
2350 CLS : CSR 30,0: IF T=0 THEN  PRINT O$ ELSE  PRINT FILE$(1)
2360 CSR 29,1: IF T=0 THEN  PRINT "==========" ELSE  PRINT "=========="
2370 RETURN
2400 LPRINT : CSR 25,5: PRINT PL$: CSR 25,7: PRINT PR$
2410 FOR I=1 TO Z: CSR 47,7: PRINT I
2420 FOR X=1 TO H: LPRINT RECORD$(I,O(X)),: CSR 0,X+14: PRINT CHR$(5): CSR 25,X+14: PRINT RECORD$(I,O(X)): NEXT X: LPRINT
2430 NEXT I
2440 CLS : RETURN
2450 IF DP=1 THEN  LPRINT CHR$(15) ELSE  LPRINT CHR$(15);CHR$(27);"H"
2460 RETURN
2500 FOR X=0 TO 79: PRINT "-";: NEXT : RETURN
2550 FOR X=0 TO 79: LPRINT "-";: NEXT : RETURN
2600 IF H=1 THEN  FOR X=1 TO 5: LPRINT : NEXT
2610 IF H=2 OR H=3 THEN  FOR X=1 TO 4: LPRINT : NEXT
2620 IF H=4 OR H=5 THEN  FOR X=1 TO 3: LPRINT : NEXT
2630 IF H=6 OR H=7 THEN  LPRINT : LPRINT
2640 IF H=8 OR H=9 THEN  LPRINT
2650 RETURN
2700 LPRINT CHR$(27);"W";CHR$(1);CHR$(27);"-";CHR$(1)
2710 IF T=1 THEN  LPRINT NAME$;" AS AT ";DATE$ ELSE  LPRINT O$;" AS AT ";DATE$
2720 LPRINT CHR$(27);"W";CHR$(0);CHR$(27);"-";CHR$(0)
2730 RETURN
2750 CLS : GOSUB 2350: CSR 25,5: PRINT "Printing completed": PAUSE 2000
2760 CSR 25,7: PRINT PL$: CSR 25,9: PRINT "Resetting computer"
2770 IF IT=1 THEN  LET IT=0: LPRINT CHR$(27);"5"
2780 IF E=1 THEN  LET E=0: LPRINT CHR$(27);"F"
2790 IF DP=1 THEN  LET DP=0: LPRINT CHR$(27);"H"
2800 PAUSE 2000: GOSUB 3200: GOTO 340
2850 REM Set Type
2860 CLS : CSR 32,0: PRINT "Print Options": CSR 31,1: PRINT "=========="
2870 CSR 10,5: PRINT "1.  Elite Pitch (96 Chars)    2.  Pica Pitch (80 chars)"
2880 CSR 26,10: INPUT "Enter 1 or 2 > ";P
2890 IF P<1 OR P)2 THEN  CSR 26,12: PRINT "Error - 1 or 2 only": PAUSE 2000: CLS : GOTO 2860
2900 IF P=1 THEN  LPRINT CHR$(27);"P";CHR$(0) ELSE  LPRINT CHR$(27);"P";CHR$(1)
2910 CLS : CSR 32,0: PRINT "Character Type": CSR 31,1: PRINT "=========="
2920 CSR 32,5: PRINT "1.  Normal"
2930 CSR 32,7: PRINT "2.  Emphasised": IF E=1 THEN  CSR 50,7: PRINT "SET"
2940 CSR 32,9: PRINT "3.  Double Print": IF DP=1 THEN  CSR 50,9: PRINT "SET"
2950 CSR 32,11: PRINT "4.  Italics": IF IT=1 THEN  CSR 50,11: PRINT "SET"
2960 CSR 32,13: PRINT "5.  All set"
```

MEMOPAD

```
2970 PRINT : GOSUB 2500
2980 PRINT : CSR 32,20: INPUT "Enter option 1 to 5 > ";OP
2990 IF OP<1 OR OP>5 THEN  CSR 0,20: PRINT CHR$(5): CSR 25,20: PRINT "Error - 1 to 5 only": PAUSE 2000: GOTO 2910
3000 ON OP-1 GOTO 3050,3010,3020,3030,3050
3010 LPRINT CHR$(27);"E": LET E=1: GOTO 3040
3020 LPRINT CHR$(27);"6": LET DP=1: GOTO 3040
3030 LPRINT CHR$(27);"4": LET IT=1
3040 CLS : GOTO 2910
3050 RETURN
3100 IF T=0 THEN  GOTO 3190
3120 CLS : CSR 20,10: PRINT "Finished with ";NAME$;"? (Y/N)": GOSUB 2050
3130 IF A$="N" OR A$="n" THEN  GOTO 3150
3140 USER ERAFILE$(1)
3145 GOTO 3190
3150 CLS : CSR 20,5: PRINT "Please wait.....": CSR 20,7: PRINT "Changing file name"
3155 USER RENNAME$=FILE$(1)
3160 USER OPEN£1,NAME$+".HDS","O"
3170 FOR X=1 TO W
3172 USER  PRINT £1,HEAD$(X)
3174 NEXT X
3180 USER CLOSE£1
3190 RETURN
3200 CLS : CSR 20,10: PRINT "F1.....Print file again": CSR 20,12: PRINT "F2.....Print another file": CSR 20,14: PRINT
"F3.....Finished with printing"
3210 LET A$=INKEY$: LET A=ASC(A$): IF A<128 OR A>130 THEN  GOTO 3210
3220 IF A=128 THEN  GOTO 510
3230 IF A=129 THEN  GOTO 80
3240 RETURN
3300 CLS : LET X=0
3310 FOR N=1 TO 15
3320 LET X=X+1: IF X>Z THEN  GOTO 3345
3330 CSR 0,N: PRINT X: CSR 5,N: PRINT RECORD$(X,1): CSR 17,N: PRINT RECORD$(X,2)
3340 NEXT N
3345 IF X>Z THEN  CSR 20,18: PRINT "E N D  O F  F I L E": LET X=0
3350 CSR 1,20: PRINT "F1.....Continue Directory    F2.....Print Record    F3.....Return to Menu"
3360 LET A$=INKEY$: LET A=ASC(A$): IF A<128 OR A>130 THEN  GOTO 3360
3370 IF A=128 THEN  CLS : GOTO 3310
3380 IF A=130 THEN  GOTO 340
3390 CLS : RETURN
3400 CSR 0,20: PRINT CHR$(5): CSR 20,20: INPUT "How many copies to make? > ";C
3410 RETURN
3450 LET H=5: GOSUB 2250
3460 FOR Q=1 TO C: GOSUB 2700
3470 CLS : LPRINT CHR$(27);"D";CHR$(1);CHR$(27);CHR$(53);CHR$(79);CHR$(105);CHR$(0)
3480 GOSUB 2350: GOSUB 2450: GOSUB 2400
3490 NEXT Q: LPRINT CHR$(27);"D";CHR$(0);CHR$(18)
3500 RETURN
3550 CLS : CSR 10,10: PRINT "Name for search list? : CSR 10,12:p.(Maximum 8 chars) > ";NAME$: IF LEN (NAME$)>8 THEN  GOTO 3550
3560 RETURN
3600 CLS : FOR X=5 TO 4+W: CSR 0,X: PRINT X-4;".": CSR 5,X: PRINT HEAD$(X-4): NEXT X
3610 CSR 20,10: INPUT "Which field to search? > ";FIELD: IF FIELD<1 OR FIELD>W THEN  GOTO 3610
3620 CSR 20,12: INPUT "Enter data to search for > ";SEARCH$: IF LEN (SEARCH$)>25 THEN  GOTO 3620 ELSE  LET SER=LEN (SEARCH$)
3640 GOSUB 4000
3650 FOR I=1 TO Z: CSR 48,14: PRINT I
3660 IF LEFT$(RECORD$(I,FIELD),SER)=SEARCH$ THEN  GOSUB 3750
3670 NEXT I
3680 FOR X=10 TO 14 STEP 2: CSR 0,X: PRINT CHR$(5): NEXT X
3690 CSR 20,10: PRINT "E N D  O F  F I L E": PAUSE 2000
```

```
3700 CSR 10,20: PRINT F$(1);"Check Directory          ";F$(2);"Return to menu"
3710 GOSUB 2000: IF A<128 OR A>129 THEN  GOTO 3710
3720 IF A=129 THEN  GOTO 340
3730 GOSUB 3300
3740 RETURN
```

```
3750 GOSUB 2350: CSR 50,0: PRINT "Record No. ";I
3760 FOR X=5 TO 4+W: CSR 20,X: PRINT X-4;".": CSR 25,X: PRINT RECORD$(I,X-4): NEXT I
3770 CSR 20,20: PRINT "Is this the record to print? (Y/N)": GOSUB 2050
3780 IF A$="Y" OR A$="y" THEN  GOSUB 4000: RETURN
3790 GOTO 790
4000 CLS : CSR 20,10: PRINT "Please wait.....": CSR 20,12: PRINT "Searching for ";
     SEARCH$: CSR 20,14: PRINT "Now searching record number "
4010 RETURN
```

Video Display Panel

```
VASAFE: PUSH AF
SAFE1:  LD A,(0FF58H)
        CP 0
        JP NZ,SAFE1
        LD A,0FFH
        LD (0FF58H),A
        POP AF
        RET
;~~~~~~~~~~~~~~~~~~~~~~~~
VAFREE: PUSH AF
        LD A,0
        LD (0FF58H),A
        POP AF
        RET
;-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
VAWRITE:;send address in HL to VDP
        PUSH AF
        LD A,L
        OUT (2),A
        LD A,H
        ADD A,40H
        OUT (2),A
        POP AF
        RET
;--------------------------------
VWRITE: OUT (1),A
        RET
;================================================================================
;Routine to increment HL in 0-3FFFH range
INCHL:  PUSH DE
        INC HL
        PUSH HL
        LD DE,ENDVRAM+1
        AND A
        SBC HL,DE
        POP HL
        POP DE
        RET C
INCHL1: LD HL,0
        RET
;--------------------------------------
;Routine to decrement HL in 0-3FFFH range
DECHL:  PUSH DE
        LD DE,1
        AND A
        SBC HL,DE
        POP DE
        RET NC
        LD HL,3FFFH
        RET
```

```
;=============================================================================
;Routine to get B number of bytes into
;data area (TABLEB) from VRAM starting
;address (HL)
DBSGET: PUSH AF          ;save registers
        PUSH DE
        LD DE,TABLEB     ;storage area
DBSGET1:PUSH HL          ;save data addr
        CALL ADDFND      ;find address of data byte
        JP NC,DBSGET2    ;data in RAM -use HL address
        POP HL           ;use HL VRAM address
        CALL VAREAD
        CALL VREAD
DBSGET3:LD (DE),A        ;save into data storage area
        INC DE
        CALL INCHL
        DJNZ DBSGET1     ;decrement count
        POP DE           ;all done
        POP AF           ;exit
        RET
DBSGET2:LD A,(HL)        ;data is in RAM
        POP HL           ;get back true address
        JP DBSGET3       ;save it
;=============================================================================
;Routine to send data in A into address
;(HL) VRAM or RAM
DBSEND: PUSH HL
        CALL ADDFND
        JP NC,DBSEND1
        POP HL
        CALL VAWRITE
        CALL VWRITE
        RET
DBSEND1:LD (HL),A
        POP HL
        RET
;=============================================================================
;Routine to find valid address of data
;save HL first, on return if C set -use HL address in RAM
;else pop HL & use VRAM address
ADDFND: PUSH DE          ;save reg
        LD DE,STRAM      ;branch of VRAM/RAM
        AND A
        SBC HL,DE
        POP DE
        RET C            ;address in VRAM ok
        PUSH DE
        LD DE,VINRAM
        ADD HL,DE
        POP DE
        AND A
        RET
;=============================================================================
;Routine to send contents of 'A' to screen
SEND:   PUSH AF          ;temp save
        PUSH AF
        SRL A            ;move top 4 bits
        SRL A            ;down
```

```
        SRL A
        SRL A
        ADD A,30H       ;add base char. no.
        CALL SENDCK
        CALL VWRITE
        POP AF
        JP SEND1
SEND2:  PUSH AF
SEND1:  AND 0FH         ;get bottom char
        ADD A,30H
        CALL SENDCK
        CALL VWRITE       ;return through VWRITE
        POP AF
        RET
SENDCK: CP 3AH
        RET C
        ADD 07H
        RET
;=============================================================================
;Routine to display address in HL to screen
;at present VRAM address followed
;by two spaces
ADDSEND:PUSH AF
        LD A,H
        CALL SEND
        LD A,L
        CALL SEND
        LD A,":"
        CALL VWRITE
        POP AF
        RET
;=============================================================================
;Routine to add 8 bytes to HL in 0-3FFFH range
ADD8:   PUSH  BC
        LD B,8
ADD81:  CALL INCHL
        DJNZ ADD81
        POP BC
        RET
;=============================================================================
;Routine to subtract 8 fram HL 0-3FFFH
SUB8:   PUSH BC
        LD B,8
SUB81:  CALL DECHL
        DJNZ SUB81
        POP BC
        RET
;=============================================================================
;Routine to store contents of A in (TABLEB)
KBSAVE: PUSH BC
        PUSH HL
        PUSH AF
        LD HL,TABLEB
        LD A,(HL)
        CP 0
        JP Z,KBSAVEX
        LD  C,A
        DEC A
```

```
          LD (HL),A
          LD B,0
          LD A,KBLEN
          SUB C
          INC A
          LD C,A
          POP AF
          ADD HL,BC
          LD (HL),A
          POP HL
          CALL VAWRITE
          CALL SEND2
          LD A,"_"
          CALL VWRITE
          INC HL
          JP KBSAVEY
KBSAVEX:  POP AF
          POP HL
KBSAVEY:  POP BC
          RET
;=============================================================================
;Routine to delete last character from (TABLEB)
KBBS:     PUSH AF
          PUSH BC
          PUSH HL
          LD HL,TABLEB
          LD A,(HL)
          CP KBLEN
          JP Z,KBBSX
          INC (HL)
          POP HL
          DEC HL
          CALL VAWRITE
          LD A,"_"
          CALL VWRITE
          LD A," "
          CALL VWRITE
          PUSH HL
KBBSX:    POP HL
          POP BC
          POP AF
          RET
;=============================================================================
;Routine to pack together values in TABLEB
;ie 03,00,0a,0f  =30AF into HL
;Default exit = HL@(BYTEST)+POINT & A set to data @ HL
HEXPACK:  PUSH IX
          PUSH BC
          PUSH DE
          LD DE,TABLEB+1   ;store pointer
          LD HL,TABLEB     ;char count store
          LD IX,TABLEB     ;data pointer
          LD A,KBLEN       ;max No. of char
          CP (HL)
          LD B,(HL)        ;count of char left
          LD (IX+0),0      ;set to 0 char to output
          JP Z,HEXIT       ;if no. data
          SUB B            ;find num in store(not spaces)
          SRL A            ;/2
```

```
         LD B,A          ;use B as  counter
         LD A,(IX+1)     ;1st data byte
         LD C,B
         JP NC,EVEN
         INC B           ;must have odd num.
         LD C,B
         JP ODD
EVEN:    INC IX          ;1st data on even count
PACK1:   SLA (IX+0)
         SLA (IX+0)
         SLA (IX+0)
         SLA (IX+0)
         LD A,(IX+1)
         ADD A,(IX+0)
ODD:     LD (DE),A
         INC (HL)
         INC DE
         INC IX
         INC IX
         DJNZ PACK1
         LD HL,TABLEB
         ADD HL,BC
         LD DE,0
         LD A,13
         LD E,(HL)
         DEC C
         JP Z,SINGLE
         DEC HL
         LD  D,(HL)
SINGLE:  PUSH DE
         POP HL
         JP HEXIT1
HEXIT:   CALL DISFULT
         LD A,13
         SCF
HEXIT1:  POP DE
         POP BC
         POP IX
         RET
;=================================================================================
;Routine to read from keyboard putting
;results in HL
;will set C if ESC,"-","." or null entery
;also a "-" will have HL set with (BYTEST)-1
;HL must hold valid address of screen to
;write to
KEYBOARD:
         PUSH DE
         LD DE,TABLEB    ;use 1st byte as buffer counter
         LD A,KBLEN      ;get buffer length
         LD (DE),A
         POP DE
         CALL VAWRITE    ;set up VRAM
READKB:  CALL 79H        ;scan Keys
         JP Z,READKB
         CP 27           ;ESC
         JP Z,EXITKB2
         CP "."          ;"." full stop
         JP Z,EXITKB2
         CP 8
```

```
                        ;back space
        JP Z,BACKSP
        CP "-"          ;"-" mimus
        JP NZ,KB1
        LD HL,(BYTEST)  ;on exit
        CALL DECHL      ;return with data @ (BYTEST)-1
EXITKB2:SCF
EXITKB: RET
KB1:    CP 13
        JP NZ,KB2
KBEXIT: CALL HEXPACK
        JP EXITKB
KB2:    AND 7FH         ;clear lower case
        CP "G"
        JP NC,READKB
KBB:    CP "A"
        JP C,KB3
        SUB 7
        JP KB4
KB3:    CP 3AH
        JP NC,READKB
        CP "0"
        JP C,READKB
KB4:    SUB 30H
        CALL KBSAVE
        JP READKB
BACKSP: CALL KBBS
        JP READKB
;=============================================================================
;Routine to display 64 bytes with address
;and cell pattern on screen
;start is at (BYTEST)-8*3
WRITETBL:
        PUSH AF
        PUSH BC
        PUSH HL
        PUSH DE
        PUSH IX
        LD HL,(BYTEST)  ;control byte
        CALL SUB8       ;start of table
        CALL SUB8
        CALL SUB8
        PUSH HL         ;save it
        LD B,64         ;8*8 table
        CALL DBSGET     ;get form VRAM add HL
        LD HL,CPATGEN   ;base add of cell's
        CALL VAWRITE    ;set it up
        LD HL,TABLEB
        LD B,64
TBL1:   LD A,(HL)       ;sent copy of data to VDP
        CALL VWRITE     ;cell pattrens
        INC HL
        DJNZ TBL1
        LD B,8          ;count
        LD A,CPAT       ;1st  pattern no.
        LD C,A
        LD HL,NAMEADD   ;place to put data
        CALL VAWRITE
        LD DE,TABLEB    ;base of data
        POP HL          ;adderss of 1st data value
```

FIRST SLICE YOUR STRING....

Most readers will have tried at some time to parse a sentence or expression or to extract information from a string of indefinite content,and will have found that it is surprisingly hard to do.

Books on advanced 'C' programming describe a very powerful function called - with the usual UNIX contempt for speech - '*strtok()' (STRing TOKeniser) which can extract words from sentences,split sentences at a word or punctuation mark,separate numbers from letters and form the basis of a lot of very clever string processing.Since it isn't actually in any of the three 'C' libraries which I have examined,I had to write it from a formal description.It was,incidentally,a **BUGGER** to write,owing to the large number of special cases.

Before I give the code for *strtok() with a simple demo program and a print-out of a test run I had best explain how it works.

Function Description:

```
char *strtok(tokens,delimiters)
char *tokens,  *delimiters;
```

Tokens is a string containing characters or groups of characters which you want ('tokens') separated by characters which you don't want (delimiters).

Strtok is called first with the actual token string as the first parameter and at least some of the delimiters in the second string.The function strips any leading delimiters,finds the first ocurrence of any of the delimiters,replaces it with a null and returns a pointer to the start of the string.

Subsequent calls are made with a null string as the first parameter.The previous token and the inserted null are removed and the search is repeated until no more tokens can be found,after which the empty string is returned.Since the source string is destroyed by the function you must first save it if you want to use it again.The delimiter string may be different with each call and may even be a token returned by a previous call.The token string does not have to be emptied before you call strtok again with a new one.

Just think - you may never need to write more than a skeleton parse routine again!

Brian Houghton

```
#include                    "stdio.h"
#ifdef  DEMO

main()
{
        extern  char    *gets(), *strtok();
        char    *p,s[100],tok[100];
        for (;;)
        {
        puts(">");
        gets (s);
        puts("Token delimiters>");
        gets (tok);

        p = strtok(s,tok);
        printf("%s",p);
        while (*p)
                {
                p = strtok("\0",tok);
                if (*p)
                        printf("!%s",p);
                };
        puts("\n");
        }
}
#endif

char    *strtok(s1,s2)
char    *s1,*s2;
{
        extern  char    *strpbrk(), *strchr();
        static  char    *p,*z;
        char    *k, *i;
        if (*s1)
                {
                p = s1;
                for (p;(i=strchr(s2,*p))!=NULL && *p;++p);
                if (*p == '\0') return (p);
                z = p;
                while (*z) z++;
                }
        else
                if (*p) {
                        for (;*p;++p);
                        if (p == z) return (p);
                        p++;
                        for (p;(i=strchr(s2,*p))!=NULL && *p;++p);
                        if (*p == '\0') return (p);
                        };
                if ((k=strpbrk(p,s2))!=NULL)
                        *k = '\0';
                return (p);
}
```

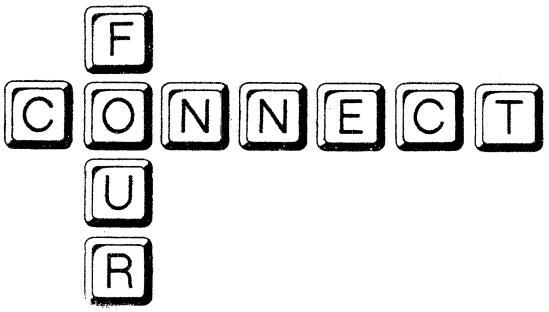
These functions should be in your library,but commercial C libraries vary a lot.*strchr() is sometimes called *index().

MEMOPAD

```
char *strpbrk(s1, s2)
  char *s1, *s2;
{
  while (*s1)
    {
      if (strchr(s2, *s1)) return s1;
      ++s1;
    }
  return NULL;
}

char *strchr(s, c)
  char *s, c;
{
  do
    {
      if (*s == c) return s;
      if ( ! *s)   return NULL;
      ++s;
    }
  while (TRUE);
}
```

```
>
this.is.the.strtok.demo
Token delimiters>
.
this!is!the!strtok!demo

>
123mumtiple98delimiters4567at12start23annd2end678888
Token delimiters>
1234567890


>
123word23word45word45
Token delimiters>
12345
word!word!word

>
123word234word345word45
Token delimiters>
wodrow
123!234!345!45

>
splitting   a string, at  the first comma.
Token delimiters>
,
splitting   a string! at  the first comma.

>
```

MEMOPAD



This month,before you start typing the supplied listing, find the lable **FORP.** Now look down the code until you find **CALL L60** (around £47EA). Enter the **edit** mode at this line by typing E **47EA** or **the address your CALL L60 appears on.**

Now add the lable **L6045: <ret>.** Your line should now look exactly like the following:

L6045: CALL L60

Now you can find the location we finished with last time and type in listing **A.**

When you have finished typing this listing go back to the assembler **prompt mode** > and go to the end of your file now type in the two subroutines **CMP & RND.**

Next month we will finish the computer move and put a **dummy** loop into the code so that you can test the code so far. We shall,the next month, almost complete the game and start debugging. I have left one bug in the code for you to get some practice. It is in a previous part of the code but the clue lies in a similar routine used in this month's code. Try and work it out ... it is the only way you will learn. If you can't **don't commit suicide** I will point it out to you in the debugging stage.

Have fun ....

```
                        TITLE Connect Four Assembler version for magazines only (c) K. Hook 1987
                        .Z80
0000'                   ASEG
                        .list
48A0    3A 4079             LD      A,(FV)
48A3    FE 01               CP      01H.
48A5    28 0E               JR      Z,L6155         ;If F=1 Then 6155
48A7    3E 01               LD      A,01H           ;6150 LET F=1
48A9    32 4079             LD      (FV),A
48AC    3A 4067             LD      A,(H$)
48AF    32 4068             LD      (X$),A          ;X$=H$
48B2    C3 47EA             JP      L6045                   ;Go check through loop again
```

```
48B5                    L6155:
48B5  3A 4078                 LD      A,(RV)
48B8  3C                      INC     A
48B9  32 4078                 LD      (RV),A
48BC  FE 09                   CP      9
48BE  30 1E                   JR      NC,L6170        ;If R>8  always test one more for NC
48C0  CD 4B5E                 CALL    L60             ;Go and evaluate

48C3  21 403A                 LD      HL,DIMA
48C6  06 01                   LD      B,01            ;FOR I= 1 TO 4

48C8                    L6165:
48C8  7E                      LD      A,(HL)
48C9  FE 04                   CP      04H             ;IF A(I)>3
48CB  38 07                   JR      C,L6166                 ;NO IT ISN'T
48CD  21 407F                 LD      HL,EV           ;THEN E=2
48D0  36 02                   LD      (HL),02H
48D2  18 07                   JR      L6170

48D4                    L6166:
48D4  23                      INC     HL              ;ELSE NEXT I
48D5  04                      INC     B
48D6  78                      LD      A,B
48D7  FE 05                   CP      05H
48D9  20 ED                   JR      NZ,L6165

48DB                    L6170:
48DB  21 407F                 LD      HL,EV           ;Point HL & DE at memory locations
48DE  11 407C                 LD      DE,UV           ;so we can compare
48E1  CD 4B48                 CALL    CMP             ;COMPARE HL,DE
48E4  30 12                   JR      NC,L6180        ;E>U
48E6  38 1C                   JR      C,L6181         ;E<U

48E8                    L6175:

48E8  21 4B24                 LD      HL,PARAMS
48EB  3E 05                   LD      A,5
48ED  77                      LD      (HL),A
48EE  CD 4AD4                 CALL    RND             ;GET A RANDOM NO
48F1  3A 4B20                 LD      A,(VAL)         ;Get actual random number
48F4  FE 03                   CP      3               ;PSEUDO routine for line 6175
48F6  28 0C                   JR      Z,L6181

48F8                    L6180:
48F8  2A 407F                 LD      HL,(EV)
48FB  22 407C                 LD      (UV),HL         ;UV=EV
48FE  3A 4072                 LD      A,(XV)
4901  32 407D                 LD      (P6),A

4904                    L6181:
4904  3A 406B                 LD      A,(TH$)         ;Get think graphic
4907  11 0609                 LD      DE,0609H        ;Colour
490A  CD 4A94                 CALL    COMSCR                  ;Go change it
490D  CD 419A                 CALL    KSUB1           ;Now print 'think' graphic
4910  3E 01                   LD      A,1
4912  32 FE14                 LD      (0FE14H),A      ;Channel
4915  3E FF                   LD      A,255
4917  32 FE16                 LD      (0FE16H),A      ;Frequency
491A  3E 0F                   LD      A,15
```

```
491C   32 FE18                  LD      (0FE18H),A          ;Volume
491F   CD 08F6                  CALL    08F6H               ;Go send sound
4922   CD 4A89                  CALL    DELAY
4925   E1                       POP     HL                  ;Get HL back from beginning of loop
4926   23                       INC     HL
4927   79                       LD      A,C
4928   3C                       INC     A
4929   4F                       LD      C,A
492A   FE 09                    CP      09H
492C   C2 47CB                  JP      NZ,FORP             ;8 times around loop

                        ;************************
                        ;Continue here next month

                        .list
                        ;
                        ;Random number routine.  Call this routine then load parameters into PARA which is
                        ;a two byte word.  Result is return in VAL  BC must be preserved until
                        ;result has been obtained. Procedure:  LD A,R:CALL
                        ;RND: LD HL,PARAMS: LD A,HIGH
                        ;VALUE: CALL PARA : LD A,(VAL) = RANDOM NUMBER
                        ;
4AD4                    RND:
4AD4   F5                       PUSH    AF
4AD5   C5                       PUSH    BC
4AD6   D5                       PUSH    DE
4AD7   E5                       PUSH    HL
4AD8   ED 5F                    LD      A,R
4ADA   32 4B1F                  LD      (SEED3),A
4ADD   ED 5B 4B1C               LD      DE,(SEED)
4AE1   2A 4B1E                  LD      HL,(SEED2)
4AE4   06 07                    LD      B,07
4AE6                    RND10:
4AE6   CD 4B04                  CALL    SHIFT
4AE9   10 FB                    DJNZ    RND10
4AEB   06 03                    LD      B,03
4AED                    RND20:
4AED   CD 4B0A                  CALL    SUB
4AF0   10 FB                    DJNZ    RND20
4AF2   ED 53 4B1C               LD      (SEED),DE

4AF6   22 4B1E                  LD      (SEED2),HL
4AF9   3E 7F                    LD      A,7FH
4AFB   A2                       AND     D
4AFC   32 4B20                  LD      (VAL),A         ;TEMP STORE FOR RANDOM NUMBER BEFORE
4AFF   E1                       POP     HL              ;CALLING PARAMS
4B00   D1                       POP     DE
4B01   C1                       POP     BC
4B02   F1                       POP     AF
4B03   C9                       RET
4B04                    SHIFT:
4B04   29                       ADD     HL,HL
4B05   EB                       EX      DE,HL
4B06   ED 6A                    ADC     HL,HL
4B08   EB                       EX      DE,HL
4B09   C9                       RET
4B0A                    SUB:
4B0A   C5                       PUSH    BC
4B0B   ED 4B 4B1E               LD      BC,(SEED2)
```

```
4B0F    B7                      OR      A
4B10    ED 42                   SBC     HL,BC
4B12    EB                      EX      DE,HL
4B13    ED 4B 4B1C              LD      BC,(SEED)
4B17    ED 42                   SBC     HL,BC
4B19    EB                      EX      DE,HL
4B1A    C1                      POP     BC
4B1B    C9                      RET
4B1C    00              SEED:   DB      00H
4B1D    00              SEED1:  DB      00H
4B1E    00              SEED2:  DB      00H
4B1F    00              SEED3:  DB      00H
4B20    00 00           VAL:    DB      00H,00H
4B22    00 00           PARA:   DB      00H,00H
                        ;
4B24                    PARAMS:
4B24    C5                      PUSH    BC
4B25    D5                      PUSH    DE
4B26    E5                      PUSH    HL
4B27    ED 5B 4B22              LD      DE,(PARA)
4B2B    3A 4B20                 LD      A,(VAL)
4B2E    6F                      LD      L,A
4B2F    26 00                   LD      H,0

4B31                    PARA1:
4B31    7D                      LD      A,L
4B32    BB                      CP      E
4B33    28 06                   JR      Z,PARA2
4B35    38 04                   JR      C,PARA2
4B37    ED 52                   SBC     HL,DE
4B39    18 F6                   JR      PARA1
4B3B                    PARA2:
4B3B    7D                      LD      A,L
4B3C    FE 00                   CP      00H
4B3E    20 01                   JR      NZ,PARA3
4B40    23                      INC     HL
4B41                    PARA3:
4B41    22 4B20                 LD      (VAL),HL

4B44    E1                      POP     HL
4B45    D1                      POP     DE
4B46    C1                      POP     BC
4B47    C9                      RET


4B48                    CMP:
                        ;Revised Compare routine V2.4
                        ;ENTRY  HL points to first operand
                        ;       DE points to second operand
                        ;EXIT   Z flag    HL=DE
                        ;       NZ        HL<> DE
                        ;       C flag    HL < DE
                        ;       NC & NZ   HL > DE
```

```
4B48    C5                      PUSH    BC
4B49    3E 03                   LD      A,3
4B4B    0E 03                   LD      C,3             ;LENGTH OF ARRAYS
4B4D    06 00                   LD      B,0
4B4F    09                      ADD     HL,BC
4B50    EB                      EX      DE,HL           ;DE = FIRST OPERAND
4B51    09                      ADD     HL,BC           ;HL=  2ND OPERAND
4B52    41                      LD      B,C             ;B=LENGTH
4B53    B7                      OR      A               ;CLEAR CARRY FLAG

4B54                    CMPLP:
4B54    2B                      DEC     HL              ;GET LESS SIG BYTE
4B55    1B                      DEC     DE              ;NOTE LESS NOT LEAST!!
4B56    1A                      LD      A,(DE)          ;GET A BYTE
4B57    9E                      SBC     A,(HL)
4B58    20 02                   JR      NZ,GOB_K                ;RET IF <> WITH FLAGS SET
4B5A    10 F8                   DJNZ    CMPLP

4B5C                    GOB_K:
4B5C    C1                      POP     BC
4B5D    C9                      RET
```

Since this is a double issue we have decided to give you next months installment of 'Connect Four' now. LUCKY YOU!!!

```
                        TITLE Connect Four Assembler version for magazines only <c> K. Hook 1987
                        .Z80
0000'                   ASEG
                        .LIST
492F    3A 407D                 LD      A,(P6)
4932    B7                      OR      A               ;Test for zero
4933    CA 4BFA                 JP      Z,DRAW          ;Its a draw!
4936    18 05                   JR      MVE             ;Otherwise go and move

4938                    L6210:
4938    E1                      POP     HL              ;Clear Stack!
4939    E1                      POP     HL              ;From JP at L6060 (JP NC,L6210)
493A    3A 4072                 LD      A,(XV)
493D                    MVE:
493D    F5                      PUSH    AF              ;Save it on stack
493E    DD 36 00 05             LD      (IX+00H),5      ;Align cursor for
4942    DD 36 01 16             LD      (IX+01H),22     ;To clear message area
4946    21 4152                 LD      HL,SPC          ;Space message
4949    CD 4A0C                 CALL    PRINT
494C    DD 36 00 05             LD      (IX+00H),5      ;Now re-align for message
4950    DD 36 01 16             LD      (IX+01H),22
4954    11 060A                 LD      DE,060AH        ;Change colour to yellow
4957    CD 4B2F                 CALL    COMSCR
495A    21 4127                 LD      HL,MYGO         ;I'm going in column x..
495D    CD 4A0C                 CALL    PRINT           ;Now print message
4960    F1                      POP     AF              ;Now get column number
4961    C6 30                   ADD     A,30H           ;Add 30h to convert to printable
```

```
4963  CD 419A          CALL   KSUB1          ;Ascii character.
4966  21 403E          LD     HL,DIMR        ;Now print it
4969  3A 4072          LD     A,(XV)
496C  3D               DEC    A              ;For array access
496D  5F               LD     E,A
496E  16 00            LD     D,0
4970  19               ADD    HL,DE          ;Now points to correct memory
4971  7E               LD     A,(HL)
4972  3C               INC    A
4973  77               LD     (HL),A         ;R(X)=R(X)+1 & R=R(X)+1
4974  32 4078          LD     (RV),A

                  ;**********************
                  ;DISPLAY COMPUTER MOVE
                  ;**********************

4977  21 400A          LD     HL,DIMGC       ;GC(x)
497A  19               ADD    HL,DE          ;DE already aligned
497B  7E               LD     A,(HL)         ;Get cursor X pos
497C  DD 77 00         LD     (IX+00H),A
497F  D5               PUSH   DE                    ;Save de for a mo
4980  3A 4078          LD     A,(RV)
4983  3D               DEC    A              ;For alignment
4984  5F               LD     E,A
4985  21 4012          LD     HL,DIMDN
4988  19               ADD    HL,DE
4989  7E               LD     A,(HL)
498A  DD 77 01         LD     (IX+01H),A              ;Now cursor Y position

                                              ;Screen postion now set up to
                                              ;display computer graphic
498D  D1               POP    DE             ;Now find memory position
498E  21 404E          LD     HL,TG$TAB      ;Pointer to addresses in array
4991  CB 23            SLA    E              ;* 2 for memory access
4993  19               ADD    HL,DE          ;D already = 0
4994  5E               LD     E,(HL)         ;Now get actual ARRAY address
4995  23               INC    HL
4996  56               LD     D,(HL)
4997  3A 4078          LD     A,(RV)
499A  3D               DEC    A
499B  6F               LD     L,A
499C  26 00            LD     H,0
499E  19               ADD    HL,DE          ;HL no points to element in array
                                              ;T$TAB(R,X)
499F  3A 4069          LD     A,(C$)         ;Computer graphic
49A2  32 4068          LD     (X$),A         ;X$=C$
49A5  77               LD     (HL),A         ;T$TAB(R,X)=C$
49A6  F5               PUSH   AF             ;Save AF Ksub1 doesn't preserve

                  ;**********************
                  ;NOW PRINT ON SCREEN
                  ;**********************

49A7  11 0608          LD     DE,0608H       ;Red for computer graphic
49AA  CD 4B2F          CALL   COMSCR
49AD  CD 419A          CALL   KSUB1                 ;Print lefthand graphic
49B0  F1               POP    AF
49B1  33               INC    AF             ;Righthand graphic
```

MEMOPAD

```
49B2    CD 419A             CALL    KSUB1
49B5    3E 01               LD      A,1             ;Channel
49B7    32 FE14             LD      (0FE14H),A
49BA    3E C8               LD      A,200           ;Frequency
49BC    32 FE16             LD      (0FE16H),A
49BF    3E 0F               LD      A,15
49C1    32 FE18             LD      (0FE18H),A      ;Volume
49C4    CD 08F6             CALL    08F6H           ;Trigger sound
49C7    CD 4BF9             CALL    L60
                    ;************************
                    ;Continue here next month
```

NewsFlash!

We are currently in the process of putting together a 'Teach Yourself Assembly Language' manual and we would appreciate some end-user input. We would like to know what specific problems you encounter in trying to learn machine code, we will then try to answer the problems within the book.

In an attempt to make it easy to follow we will relate instructions to a Basic statements (where possible). The book will retails at around £14.95 and will also contain each Z80 instruction with an explanation of what it does etc. in a way that hasn't been used before.

Please don't be afraid to put your problems on paper - however trivial you may think they are - all letters are confidential and I am sure you are not the only user who has come unstuck in the same area.

The book will be titles "The Black Book of Assembly Language".

THE SOURCE

WE STILL HAVE A LIMITED SUPPLY OF THE 2nd PRINT

7.95p

CONTENTS INCLUDE.....
How to configure the VDP for graphics
Sprite detection .. Sound .. RST10
complete with demonstrations.....

THE DISC OF PROGRAMS FROM *THE SOURCE* NOW AVAILABLE ....*LET YOUR FINGERS HAVE A REST* .....*GET IT RIGHT FIRST TIME !* **9.95p**

D.I.Y

INVOICING

Print your own invoices and address labels as well as produce a print out of all your customers with their corresponding customer number for easy reference. The program works quickly and the on screen prompts make it easy to use.

```
0 GOTO 7
5 CODE

400E        RET
400F        JP MASBLK
4012 MAXL:  DB 0,0
4014 STR1:  DS 40
403C NOCH:  DB 0,0
403E OLDCH: DB 0,0
4040 FIN:   LD A,(NOCH)
4043        LD B,0
4045        LD C,A
4046        RET
4047 CSR:   RST 10
4048        DB 883
4049        DB 3
404A CSRX:  DB 0
404B CSRY:  DB 0
404C        RET
404D CHOK:  CP 31
404F        RET NC
4050        CP 13
4052        RET Z
4053        CP 10
4055        RET Z
4056        CP 8
4058        RET Z
4059        LD A,255
405B        RET
405C ADV:   PUSH AF
405D        LD A,(NOCH)
4060        INC A
4061        LD (NOCH),A
4064        LD A,(CSRX)
4067        INC A
4068        LD (CSRX),A
406B        POP AF
406C        RET
406D WRTA:  LD (LAB2),A
4070        RST 10
4071        DB 881
4072 LAB2:  DB 0
4073        RET
4074 CLEAR: LD A,255
4076        LD (OLDCH),A
4079        LD HL,STR1
407C        LD A,0
407E        LD (NOCH),A
4081 BLK2:  LD (HL),0
4083        INC HL
4084        INC A
4085        CP 30
4087        RET Z
4088        JP BLK2
408B CHLIN: LD A,(NOCH)
408E        LD D,A
408F        LD A,(MAXL)
4092        DEC A
4093        CP D
4094        CALL NZ,ADV
4097        RET
4098 STDT:  LD HL,STR1
409B        PUSH HL
409C        POP DE
409D        LD HL,(NOCH)
40A0        ADD HL,DE
40A1        LD (HL),A
40A2        RET
40A3 DEL:   LD A,(NOCH)
40A6        CP 0
40A8        JR Z,BLK1
40AA        DEC A
40AB        LD (NOCH),A
40AE        LD A,(CSRX)
40B1        DEC A
40B2        LD (CSRX),A
40B5        LD A,8FF
40B7        LD (OLDCH),A
40BA        CALL CSR
40BD        CALL PCSR
40C0        JR BLK1
40C2 PCSR:  CALL CSR
40C5        LD A,(NOCH)
40C8        LD D,A
40C9        LD A,(MAXL)
40CC        CP D
40CD        RET Z
40CE        LD A,(OLDCH)
40D1        CP D
40D2        RET Z
40D3        LD A,"+"
40D5        CALL WRTA
40D8        LD A,(MAXL)
40DB        DEC A
40DC        CP D
40DD        RET Z
40DE        LD A,32
40E0        CALL WRTA
40E3        RET
40E4 FIN1:  LD A,(MAXL)
40E7        LD D,A
40E8        LD A,13
40EA        CALL WRTA
40ED        LD A,(NOCH)
40F0        INC A
40F1        CP D
40F2        JP Z,FIN
40F5        DEC A
40F6        LD (NOCH),A
40F9        CALL CSR
40FC        LD A,32
40FE        CALL WRTA
4101        JP FIN
4104 MASBLK: CALL CLEAR
4107        CALL CSR
410A        JP LAB3
410D BLK1:  CALL 879
4110        JR Z,BLK1
4112        CALL CHOK
4115        CP 255
```

```
4117        JR Z,BLK1
4119        CP 13
411B        JR Z,FIN1
411D        CP 10
411F        JR Z,FIN1
4121        CP 8
4123        JP Z,DEL
4126        CALL CSR
4129        CALL WRTA
412C        CALL STDT
412F        LD A,(NOCH)
4132        LD (OLDCH),A
4135        CALL CHLIM
4138 LAB3:  CALL PCSR
4138        JP BLK1
413E        RET

Symbols:
MASBLK  4104    MAXL    4012
STR1    4014    NOCH    403C
OLDCH   403E    FIN     4040
CSR     4047    CSRX    404A
CSRY    404B    CHOK    404D
ADV     405C    LAB2    4072
WRTA    406D    CLEAR   4074
BLK2    4081    CHLIM   408B
STDT    4098    DEL     40A3
BLK1    410D    PCSR    40C2
FIN1    40E4    LAB3    4138
```

```
7 POKE 64145,128: POKE 64862,15
10 CLS : CSR 1,2: PRINT "ENTER THE No OF CUSTOMERS REQUIRED": CSR 1,4: PRINT "MAXIMUM 100"
12 LET X=36: LET Y=2: LET LX=3: GOSUB 30000
15 LET MAX=VAL(A$): IF MAX<1 OR MAX>100 THEN  GOTO 12
20 CLS : CSR 2,2: PRINT "PLEASE WAIT"
50 DIM CO$(7,20),CSRX(7),CSRY(7),PX(5),PY(5),INF$(5,27),CF$(MAX,6,20),INV$(5,6),CSRL(5),CXF(5),CYF(5),CFL(5)
52 DIM CFX(6),CFY(6),DUM$(40),NC$(5,20),ACCNO$(3),PR$(10,20),QUANT(10,4),DAM$(10,4,10),VAT(10),INVT$(11)
60 FOR A=1 TO 7
62 READ CSRX(A): READ CSRY(A)
64 NEXT A
65 FOR A=1 TO 5
66 READ PX(A): READ PY(A)
67 FOR P=1 TO 20: LET NC$(A,P)=" ": NEXT
68 LET PR$(A)="                    ": NEXT
70 FOR A=1 TO 5
71 LET PR$(A+5)="                    "
72 READ INF$(A)
74 NEXT
80 FOR P=1 TO 6: READ CFX(P): READ CFY(P): NEXT
85 FOR P=1 TO 5: READ CSRL(P): NEXT
90 FOR P=1 TO 4: READ CXF(P),CYF(P),CFL(P): NEXT
280 FOR A=1 TO MAX: LET CF$(A,1,1)=" ": NEXT
300 FOR A=1 TO 7: LET CO$(A)="                    ": NEXT A
310 LET DUM$="                                    "
320 FOR P=1 TO 10: FOR O=1 TO 4: LET DAM$(P,O)="          ": NEXT : NEXT
450 CLS : PLOB "PRO1"
500 IF INKEY$="" THEN  GOTO 500 ELSE  LET KP=ASC(INKEY$)
510 LET KP=KP-127
```

```
515 IF KP<1 OR KP>4 THEN  GOTO 500
520 ON KP GOTO 500,1000,2000,3000,4000
525 GOTO 500
1000 REM
1002 PLOD "PR02"
1010 FOR A=1 TO 7
1012 LET X=CSRX(A): LET Y=CSRY(A): LET LX=20: GOSUB 30000
1020 LET CO$(A)=A$
1030 NEXT
1032 IF LEFT$(CO$(1),4)="QUIT" THEN  POKE 64145,128: POKE 64862,15
1035 CSR 2,19: PRINT CHR$(5);"PRESS <ESC> TO RETURN TO MAIN MENU"
1040 GOSUB 9000: IF Q<>27 THEN  GOTO 1040 ELSE  GOTO 450
2000 CLS : PLOD "PR099"
2005 GOSUB 9000
2006 IF Q=13 THEN  GOTO 2010
2007 IF Q=27 THEN  GOTO 450
2008 GOTO 2005
2010 SAVE "INVOICE.PTR"
2020 GOTO 450
3000 REM
3010 PLOD "PR03"
3020 GOSUB 9000
3030 IF Q=128 THEN  GOTO 3100
3035 IF Q=129 THEN  GOTO 3500
3037 IF Q=130 THEN  GOTO 3600
3040 IF Q=27 THEN  GOTO 450
3050 GOTO 3020
3100 REM
3110 CLS : PLOD "PR04"
3120 FOR A=1 TO 5
3122 CSR 2,20: PRINT CHR$(5);INF$(A)
3125 LET X=PX(A): LET Y=PY(A): LET LX=CSRL(A): GOSUB 30000
3130 LET INV$(A)=A$
3131 IF A=1 THEN  GOTO 8000
3132 IF A=3 THEN  GOTO 8010
3135 NEXT
3161 LET LL=VAL(INV$(3))
3162 FOR INO=1 TO LL
3163 PLOD "PR08"
3165 CSR 28,16: PRINT INO
3166 FOR P=1 TO 4
3168 LET X=CXF(P): LET Y=CYF(P): LET LX=CFL(P): GOSUB 30000: IF P=2 THEN  GOTO 3172 ELSE  IF VAL(A$)=0 THEN  GOTO 3168
3170 IF P=4 THEN  LET QUANT(INO,P)=INT(VAL(A$)*100)/100: LET DAN$(INO,P)=STR$(QUANT(INO,P))
3172 IF P=2 THEN  LET PR$(INO)=A$
3173 IF P=1 THEN  LET QUANT(INO,P)=INT(VAL(A$)*100)/100: LET DAN$(INO,P)=STR$(QUANT(INO,P))
3175 IF P=3 THEN  LET QUANT(INO,P)=INT(VAL(A$)*100)/100: LET DAN$(INO,P)=STR$(QUANT(INO,P))
3176 NEXT : LET QUANT(INO,2)=INT(QUANT(INO,1)/QUANT(INO,4)*QUANT(INO,3)*100)/100: LET DAN$(INO,2)=STR$(QUANT(INO,2))
3177 CSR 3,20: PRINT "PLUS VAT Y or N": LET I=21: LET Y=20: LET LX=1: GOSUB 30000: IF A$="Y" OR A$="y" THEN  LET VAT(INO)=INT
   (QUANT(INO,2)*15)/100: GOTO 3179
3178 IF A$="N" OR A$="n" THEN  GOTO 3179 ELSE  GOTO 3177
3179 NEXT INO
3180 CLS : CSR 2,2: PRINT "IS THE CUSTOMERS ADDRESS ON FILE": CSR 2,4: PRINT "PLEASE PRESS Y OR N"
3181 GOSUB 9000: IF Q=78 THEN  GOTO 11000
3182 IF Q<>89 THEN  GOTO 3181
3183 CSR 2,10: PRINT CHR$(5): CSR 2,12: PRINT CHR$(5): CSR 2,14: PRINT CHR$(5): CSR 2,4: PRINT CHR$(5): CSR 2,4: PRINT CHR$(5)
3184 CSR 2,2: PRINT "ENTER CUSTOMER NO": LET X=21: LET Y=4: LET LX=3: GOSUB 30000
3185 LET Z=VAL(A$): IF Z<1 OR Z>MAX THEN  GOTO 3183
3188 IF CF$(Z,1,1)=" " THEN  CSR 2,10: PRINT "ACCOUNT No. NOT USED": PAUSE 1000: CSR 2,10: PRINT CHR$(5): GOTO 3184
3189 CSR 2,10: PRINT CF$(Z,1)
3190 CSR 2,12: PRINT "IS THIS THE CORRECT ACCOUNT"
```

```
3191 CSR 2,14: PRINT "PLEASE PRESS Y OR N"
3192 GOSUB 9000
3193 IF Q<>89 THEN  GOTO 3183
3195 FOR P=1 TO 5: LET NC$(P)=CF$(I,P): NEXT
3197 LET ACCNO$=STR$(I)
3200 IF LEFT$(INV$(1),1)="I" OR LEFT$(INV$(1),1)="i" THEN  LET INVT$="INVOICE    " ELSE  LET INVT$="CREDIT NOTE"
3205 LET NT=0: LET TT=0: LET VT=0
3206 FOR P=1 TO VAL(INV$(3))
3207 LET NT=NT+(QUANT(P,2)): LET VT=VT+(VAT(P))
3208 NEXT
3209 LET TT=TT+NT+VT
3210 GOSUB 6000
3215 GOSUB 5000
3220 PRINT : PRINT "                                ";INVT$
3230 PRINT : PRINT : PRINT "      TO                                          FROM"
3232 PRINT
3235 FOR P=1 TO 5: PRINT "    ";NC$(P);"                          ";CO$(P): NEXT
3237 PRINT : PRINT : PRINT
3240 PRINT "         DATE     ";LEFT$(INVT$,7);"   ACCOUNT NO    YOUR REF. NO"
3245 PRINT "         ";INV$(2);"    ";INV$(5);"        ";ACCNO$;"          ";INV$(4)
3247 PRINT : PRINT : PRINT "       QUANTITY      DESCRIPTION          PRICE     PER      NETT"
3250 FOR P=1 TO VAL(INV$(3))
3255 PRINT : PRINT "         ";DAM$(P,1);"   ";PR$(P);"  ";DAM$(P,3);DAM$(P,4);DAM$(P,2)
3260 NEXT
3261 PRINT : PRINT : PRINT "                                                              NET £";NT
3262 PRINT : PRINT "                                                            VAT £";VT
3263 PRINT : PRINT "                                                           TOTAL £";TT
3265 FOR P=1 TO 30-VAL(INV$(3))
3270 PRINT : NEXT
3275 PRINT "     COMPANY REGISTRATION No.";CO$(6);"  VAT No.";CO$(7)
3276 POKE 64886,0: CLS : CSR 2,2: PRINT "ANOTHER COPY Y OR N"
3277 GOSUB 9000
3278 IF Q=78 THEN  GOTO 3280
3279 IF Q<>89 THEN  GOTO 3277 ELSE  GOTO 3215
3280 FOR P=1 TO 10: LET VAT(P)=0: NEXT
3282 FOR P=1 TO 10
3283 LET PR$(P)="                    "
3284 FOR I=1 TO 4
3286 LET DAM$(P,I)="          "
3288 NEXT
3290 NEXT
3300 POKE 64886,0: GOTO 3000
3500 GOSUB 5000
3520 PRINT : PRINT CO$(1): PRINT : PRINT "CUSTOMER LIST BY NO AND NAME"
3525 PRINT : PRINT
3530 FOR NOA=1 TO MAX
3535 IF CF$(NOA,1,1)=" " THEN  GOTO 3545
3540 PRINT NOA;"  ";CF$(NOA,1)
3545 NEXT
3550 POKE 64886,0
3560 GOTO 450
3600 GOSUB 5000
3610 FOR P=1 TO MAX STEP 3
3611 IF P=MAX THEN  GOTO 3660
3612 IF P+1=MAX THEN  GOTO 3680
3615 IF CF$(P+1,1,1)=" " THEN  GOTO 3660
3617 IF CF$(P+2,1,1)=" " THEN  GOTO 3680
3620 FOR T=1 TO 6
```

```
3622 PRINT CF$(P,T);"        ";CF$(P+1,T);"          ";CF$(P+2,T)
3624 NEXT T
3625 PRINT : PRINT : PRINT
3626 NEXT P
3630 POKE 64886,0: GOTO 3000
3660 FOR T=1 TO 6
3662 PRINT CF$(P,T)
3664 NEXT T
3666 POKE 64886,0: GOTO 3000
3680 FOR T=1 TO 6
3682 PRINT CF$(P,T);"         ";CF$(P+1,T)
3684 NEXT T
3686 POKE 64886,0: GOTO 3000
4000 REM
4010 PLOD "PRO5"
4012 GOSUB 9000
4015 IF Q=27 THEN  GOTO 450
4017 IF Q=128 THEN  GOTO 4100
4019 IF Q=129 THEN  GOTO 4300
4021 IF Q=130 THEN  GOTO 4500
4025 GOTO 4012
4100 REM
4115 FOR A=1 TO MAX: IF ASC(CF$(A,1,1))=32 THEN  GOTO 4120 ELSE  NEXT
4117 CLS : CSR 1,1: PRINT "CUSTOMER FILE FULL": PAUSE 3000: GOTO 4000
4120 LET CUST=A: CLS : PLOD "PRO6"
4122 FOR P=1 TO 6: LET CF$(A,P)="                    ": NEXT
4125 CSR 34,0: PRINT CUST: CSR 34,0: PRINT "["
4127 FOR P=1 TO 6
4129 LET X=CFX(P): LET Y=CFY(P): LET LX=20: GOSUB 30000
4130 LET CF$(A,P)=A$
4132 NEXT
4135 PAUSE 200: GOTO 4000
4300 REM
4305 CLS : CSR 2,2: PRINT "ENTER CUSTOMER NO": CSR 2,4: PRINT "RETURN ONLY TO MENU"
4307 LET X=21: LET Y=2: LET LX=3: GOSUB 30000: LET A=VAL(A$): IF A=0 THEN  GOTO 4000
4308 IF A<1 OR A>MAX THEN  GOTO 4307
4309 IF CF$(A,1,1)=" " THEN  CSR 2,10: PRINT "NOT OPENED YET": PAUSE 1000: CSR 2,10: PRINT CHR$(5): GOTO 4307
4310 CLS : PLOD "PRO7"
4312 CSR 26,0: PRINT A
4313 FOR P=1 TO 6: CSR CFX(P),CFY(P): PRINT CF$(A,P): NEXT
4320 CSR 2,20: PRINT "ENTER NO TO CHANGE OR RET TO EXIT"
4322 LET X=36: LET Y=20: LET LX=1: GOSUB 30000
4323 IF VAL(A$)<1 THEN  GOTO 4000 ELSE  IF VAL(A$)>6 THEN  GOTO 4322 ELSE  LET L=VAL(A$)
4324 LET CF$(A,L)="                    "
4325 LET X=CFX(A): LET Y=CFY(L): LET LX=20: GOSUB 30000
4345 LET CF$(A,L)=A$
4350 CSR 2,20: PRINT CHR$(5);"ANOTHER LINE Y OR N"
4351 GOSUB 9000
4353 IF Q=89 THEN  GOTO 4320
4355 IF Q<>78 THEN  GOTO 4351
4357 PAUSE 100: GOTO 4000
4499 GOTO 450
4500 REM
4510 CLS
4512 CSR 2,2: PRINT "ENTER CUSTOMER No TO BE DELETED"
4513 LET X=34: LET Y=2: LET LX=3: GOSUB 30000
4515 LET A=VAL(A$): IF A<1 OR A>MAX THEN  GOTO 4513
4517 IF CF$(A,1,1)=" " THEN  GOTO 4513
4520 FOR C=1 TO 6: LET CF$(A,C)="                ": NEXT
```

MEMOPAD

```
4530 LET CF$(A,1,1)=" "
4550 CSR 2,4: PRINT "DELETED CUSTOMER No.";A
4551 PAUSE 1000: GOTO 4000
5000 REM
5010 CLS : CSR 2,2: PRINT "PRESS ESC WHEN PRINTER IS READY": CSR 2,4: PRINT "PLEASE LINE UP THE PAPER"
5020 GOSUB 9000
5030 IF Q<>27 THEN  GOTO 5020 ELSE  POKE 64886,1: RETURN
6000 FOR P=2 TO 5
6005 FOR Q=1 TO 6
6010 IF ASC(INV$(P,Q))=0 THEN  GOSUB 6100
6015 NEXT Q
6020 NEXT P
6050 RETURN
6100 FOR T=Q TO 6
6105 LET INV$(P,T)=" "
6110 NEXT
6120 LET Q=6: RETURN
8000 IF A$="I" OR A$="C" THEN  GOTO 3135 ELSE  GOTO 3125
8010 IF VAL(A$)<1 OR VAL(A$)>10 THEN  GOTO 3125
8020 GOTO 3135
9000 IF INKEY$="" THEN  GOTO 9000 ELSE  LET Q=ASC(INKEY$)
9010 RETURN
10000 DATA 14,3,14,5,14,6,14,7,14,8,14,11,14,13
10010 DATA    30,3,30,5,30,7,30,9,23,11
10020 DATA  PLEASE ENTER "I" OR "C"
10022 DATA PLEASE ENTER DATE AS 260761
10024 DATA  MAXIMUM No 10 ITEMS
10026 DATA  ORDER REFERENCE
10028 DATA PLEASE ENTER INVOICE NO
10030 DATA 12,3,12,5,12,6,12,7,12,8,12,9
10040 DATA 1,6,2,6,6
10050 DATA 16,4,9,16,6,20,16,8,9,16,10,9
11000 FOR P=1 TO 5: LET NC$(P)="                    ": NEXT
11005 CLS : PLOD "PRO10"
11010 FOR P=1 TO 5: LET X=CSRX(P)-2: LET Y=CSRY(P): LET LX=20: GOSUB 30000
11020 LET NC$(P)=A$: NEXT
11030 LET ACCNO$="   ": GOTO 3200
30000 CSR X,Y: FOR N=1 TO LX: PRINT " ";: NEXT N
30010 POKE 16458,X: POKE 16459,Y: POKE 16402,LX: LET LX=USR(16399)
30020 LET A$="                                    "
30030 FOR N=1 TO LX+1
30040 LET A$(N)=CHR$(PEEK(16403+N))
30050 NEXT
30060 LET A$=LEFT$(A$,LX+1)
30070 RETURN
```


FOR SALE.....

SHOP SOILED TANDATA 110 MODEM

FOR PRESTEL COMMUNICATION **69.00 PLUS 5.00 POSTAGE & PACKING**

TRANSMITS AT 1200

CONTACT ORION SOFTWARE'S SALES
DEPARTMENT FOR FURTHER DETAILS

SPACE MISSION

```
10 GOTO 5500
500 SBUF 2: SOUND 3,6,0: SOUND 2,200,0: SOUND 1,300,0
600 CODE

4031            XOR A
4032            LD (£FD5E),A
4035            LD (64840),A
4038            LD A,(£FAD2)
403B            PUSH AF
403C            LD IX,£FB4B
4040 START:     LD A,£F7
4042            OUT (5),A
4044            IN A,(5)
4046            BIT 7,A
4048            JR NZ,C1
404A            LD A,(IX+0)
404D            LD C,(IX+5)
4050            SUB C
4051            LD (IX+0),A
4054            JR NC,C2
4056            DEC (IX+1)
4059 C2:        DEC (IX+2)
405C            RST 10
405D            DB £33,£21,£21,£24,£33,£21,£22,£04
4065 C1:        LD A,£EF
4067            OUT (5),A
4069            IN A,(5)
406B            BIT 7,A
406D            JR NZ,C3
406F            LD A,(IX+0)
4072            LD C,(IX+5)
4075            SUB C
4076            LD (IX+0),A
4079            JR NC,C4
407B            DEC (IX+1)
407E C4:        INC (IX+2)
4081            RST 10
4082            DB £33,£21,£21,£20,£33,£21,£22,£00
408A C3:        LD A,£DF
408C            OUT (5),A
408E            IN A,(5)
4090            BIT 7,A
4092            JR NZ,C5
4094            LD A,(IX+0)
```

```
4097          LD C,(IX+6)
409A          SUB C
409B          LD (IX+0),A
409E          JR NC,C6
40A0          DEC (IX+1)
40A3 C6:      LD A,(IX+3)
40A6          LD C,(IX+8)
40A9          ADD A,C
40AA          LD (IX+3),A
40AD          JR NC,C9
40AF          INC (IX+4)
40B2 C9:      LD A,R
40B4          AND 3
40B6          ADD A,8
40B8          LD H,A
40B9          LD A,R
40BB          AND 4
40BD          ADD A,8
40BF          LD L,A
40C0          LD (£FE63),HL
40C3          LD A,£F3
40C5          OUT (6),A
40C7          IN A,(3)
40C9          JR C7
40CB STORE:   DB £F8,2,1,£F9,3,3,£FA,4,4,£FB,5,5,£FC,5,5,£FD,4,4,£FE,3,3,£FF,2,1
40E3 C5:      LD A,£FF
40E5          OUT (6),A
40E7          IN A,(3)
40E9 C7:      LD A,(IX+3)
40EC          LD C,(IX+7)
40EF          SUB C
40F0          LD (IX+3),A
40F3          JR NC,C8
40F5          DEC (IX+4)
40F8 C8:      LD H,(IX+4)
40FB          LD L,A
40FC          SRL H
40FE          RR L
4100          SRL H
4102          RR L
4104          SRL H
4106          RR L
4108          LD (IX+9),L
410B          RST 10
410C          DB £31,£23,£01
410F          LD A,(IX+9)
4112          RST 28
4113          DB £AC
4114          RST 10
4115          DB £31,£23,£02
4118          LD A,(IX+9)
411B          RST 28
411C          DB £AC
411D          LD B,8
411F          LD HL,STORE
4122 LOOP:    LD A,£1B
4124          CALL £BC
4127          LD A,"C"
4129          CALL £BC
```

```
412C          LD A,(IX+2)
412F          LD C,(HL)
4130          ADD A,C
4131          CALL £BC
4134          INC HL
4135          LD A,(IX+9)
4138          LD C,(HL)
4139          ADD A,C
413A          CALL £BC
413D          INC HL
413E          LD A,(HL)
413F          CALL £BC
4142          LD A,(£FE1A)
4145          AND A
4146          JP NZ,END
4149          INC HL
414A          DJNZ LOOP
414C          LD A,£DF
414E          OUT (6),A
4150          IN A,(3)
4152          CALL £CB
4155          CALL £3CF1
4158          CALL £3D21
415B          LD A,(£FD49)
415E          LD B,1
4160          LD E,A
4161 PAUSE:   DJNZ PAUSE
4163          LD A,127
4165          OUT (5),A
4167          IN A,(6)
4169          AND 1
416B          CALL Z,HOLD
416E          DEC E
416F          JR NZ,PAUSE
4171          RST 10
4172          DB £33,£22,£22,£05
4176          LD A,(£FE54)
4179          BIT 5,A
417B          JR NZ,SUBEND
417D          RST 10
417E          DB £23,£27,0
4181          BIT 7,(IX+1)
4185          JR NZ,NONE
4187          LD HL,(£FB4B)
418A          LD A,H
418B          AND A
418C          JR NZ,CE
418E          LD A,£D0
4190          OUT (6),A
4192          IN A,(3)
4194 CE:      LD DE,10
4197          PUSH DE
4198          LD E,100
419A          PUSH DE
419B          LD DE,1000
419E          PUSH DE
419F          LD DE,10000
41A2          PUSH DE
41A3          LD B,4
41A5 LP2:     LD A,£2F
41A7          POP DE
41A8 LP3:     INC A
41A9          AND A
41AA          SBC HL,DE
41AC          JP P,LP3
41AF          ADD HL,DE
41B0          CALL £BC
41B3          DJNZ LP2
41B5          LD A,L
41B6          ADD A,£30
41B8          CALL £BC
41BB          JP START
41BE NONE:    LD A,£B0
41C0          OUT (6),A
41C2          IN A,(3)
41C4          RST 10
41C5          DB £30,£20,£28,£30,£21,£21,£86," ZERO "
41D2          JP C5
41D5 SUBEND:  LD A,1
41D7          LD (64840),A
41DA END:     POP AF
41DB          LD (£FAD2),A
41DE          OUT (0),A
41E0          LD A,15
41E2          LD (£FD5E),A
41E5          RET
41E6 HOLD:    LD A,191
41E8          OUT (5),A
41EA          IN A,(5)
41EC          AND 65
41EE          XOR 65
41F0          RET NZ
41F1          LD A,253
41F3          OUT (5),A
41F5          IN A,(5)
41F7          AND 1
41F9          JR NZ,HOLD
41FB          POP HL
41FC          LD HL,£FFFF
41FF          LD (HL),L
4200          LD HL,£FAD8
4203          LD (£FB41),HL
4206          LD HL,0
4209          LD (£FB43),HL
420C          JR END
420E          RET

Symbols:
C1        4065    C2        4059
C3        408A    C4        407E
C5        40E3    C6        40A3
C7        40E9    C8        40F8
C9        40B2    STORE     40CB
LOOP      4122    START     4040
LP2       41A5    LP3       41A8
NONE      41BE    CE        4194
PAUSE     4161    END       41DA
SUBEND    41D5    HOLD      41E6
```

MEMOPAD

```
610 IF PEEK(65535)=255 THEN  GOTO 5520
620 LET X=PEEK(64333): LET Y=PEEK(64340)
630 IF PEEK(64840)<>0 THEN  GOTO 2000 ELSE  GOSUB 2100
640 GOSUB 5330: GOTO 500
999 REM ** Spaceship Materialisation **
1000 POKE 64862,5: SBUF 2: SOUND 2,2200,0,-7,0,300,1: SOUND 3,3,15: COLOUR 3,13: GOSUB 1500
1010 SOUND 2,0,0,7,0,300,0: ATTR 2,1: GOSUB 1500: ATTR 2,0: SOUND 3,0,0
1020 POKE 64862,31: SPRITE 1,0,X,Y,0,0,0: SPRITE 2,2,X,Y,0,0,0: SPRITE 3,5,0,0,0,0,0: ADJSPR 3,3,208
1030 SBUF 2: SOUND 3,0,0: SOUND 2,4096,128,4090,8,112,1: FOR Q=1 TO 4: FOR A=2 TO 15: ADJSPR 1,1,A: PAUSE 20: NEXT : NEXT : ADJSPR 1,1,9
1040 RETURN
1500 COLOUR 2,0
1510 FOR A=1 TO 10: PLOT X-4,Y+4: ANGLE PI/2: DRAW A: PHI PI/2: DRAW A/2
1520 FOR B=1 TO 6: PHI PI/3: DRAW A: NEXT : NEXT
1530 RETURN
1999 REM ** Results of Collisions **
2000 POKE 64862,0: SOUND 2,60,0: SOUND 1,0,0
2010 FOR S=1 TO 16: SPRITE S,1,X,Y,S(S,1),S(S,2),9: NEXT : POKE 64862,15
2015 FOR S=15 TO -.1 STEP -1: COLOUR 4,11: PAUSE 60: COLOUR 4,0: SOUND 3,7,S: PAUSE 60: NEXT
2020 PAUSE 2000: LET SH=SH-1: IF SH=0 THEN  GOTO 6100 ELSE  POKE 64862,0: LET K=0: NEXT LE
2100 IF K=0 AND X>124 AND X<140 AND Y>91 AND Y<100 THEN  GOTO 2500
2110 POKE 64323,PEEK(64323)-1: LET A=PEEK(64321)+256*PEEK(64322)-3: POKE 64321,MOD(A,256): POKE 64322,INT(A/256)
2120 IF Y>7 THEN  GOTO 2000
2130 ADJSPR 1,2,0: ADJSPR 3,1,7: POKE 64862,15: SOUND 3,0,0
2140 FOR A=0 TO 500: SOUND 0,A,10: SOUND 1,A+50,10: SOUND 2,A+100,10: NEXT : SOUND 0,0,0: SOUND 1,0,0: SOUND 2,0,0
2150 LET S=PEEK(64331)+256*PEEK(64332): IF S<32768 AND K=1 THEN  LET SC=SC+S
2160 NEXT
2500 LET Y=97: ADJSPR 3,1,Y: ADJSPR 3,2,Y: ATTR 2,1: SOUND 3,0,0: FOR A=0 TO 16 STEP .5: COLOUR 4,A: IF PEEK(64840)<>0 THEN  ATTR 2,0: GOTO 2110
2510 SOUND 2,1000-A*60,A: LINE 120+A,94,120+A,97: NEXT : ATTR 2,0
2520 LET K=1: RETURN
2999 REM ** Draw screens **
3000 COLOUR 3,2: LINE 8,0,254,0: LINE 255,0,255,183: LINE 255,183,8,183: LINE 7,183,7,0: LINE 7,10,7,10+RND*10
3005 LET Q=2+INT(RND*27): FOR A=15 TO Q*8-9 STEP 8: LINE PEEK(65090),PEEK(65091),A,15+RND*10: NEXT : LINE PEEK(65090),PEEK(65091),Q*8-1,7
3010 COLOUR 1,11: CSR Q,23: PRINT CHR$(154);CHR$(154);CHR$(154);: CSR Q,22: PRINT CHR$(130);" ";CHR$(130): CSR Q,21: PRINT "   ": POKE 65090,Q*8+24
3015 FOR A=Q*8+31 TO 255 STEP 8: LINE PEEK(65090),PEEK(65091),A,15+RND*10: NEXT
3020 COLOUR 3,15: FOR A=1 TO N: LET X=RND*221+21: LET Y=RND*130+33: ON L GOSUB 3100,3200,3300,3400,3500,3600,3700: NEXT
3030 FOR A=4 TO 7: COLOUR 3,A: LINE 120,90+A,135,90+A: NEXT
3040 ATTR 2,1: FOR A=98 TO 105: LINE 120,A,135,A: NEXT : ATTR 2,0
3050 RETURN
3090 REM ** Information for screens **
3100 LINE X+RND*24-12,Y+RND*24-12,X+RND*24-12,Y+RND*24-12: RETURN
3110 LET G=.75: LET T=1.5: LET M=1: LET P=5: LET N=115: RETURN
3200 INK 2+RND*14: LET Q=RND*5+10: PLOT Q/2+X,Y: ANGLE PI/2: FOR QQ=1 TO 5: DRAW Q: PHI PI*.8: NEXT : RETURN
3210 LET G=2: LET T=2.5: LET M=1: LET P=10: LET N=50: RETURN
3300 IF A<N THEN  COLOUR 3,14+RND*2: PLOT X,Y: PLOT X+1,Y: PLOT X-1,Y: PLOT X,Y+1: PLOT X,Y-1: RETURN
3303 COLOUR 3,10: FOR X=21 TO 25: CIRCLE X,150,10: NEXT
3306 COLOUR 3,15: ATTR 2,1: FOR X=24 TO 37: LINE X,160,X,140: NEXT : ATTR 2,0: RETURN
3310 LET G=3: LET T=2.5: LET M=3: LET P=1: LET N=100: RETURN
3400 COLOUR 3,10+RND*6: LINE X+2,Y+2,X-3,Y-2: LINE X-3,Y+2,X+2,Y-2: RETURN
3410 LET G=2: LET T=4: LET M=1: LET P=5: LET N=100: RETURN
3420 FOR S=0 TO 14: LET Q=-RND*20-3: SPRITE S+3,4,500,26+S*10,Q,0,6: SPRITE S+18,4,1000,26+S*10,Q,0,6: NEXT : RETURN
3500 COLOUR 3,7: LINE X,Y-8,X,Y+RND*12: RETURN
3510 LET G=1: LET T=3: LET M=1: LET P=3: LET N=120: RETURN
3520 FOR S=3 TO 32: SPRITE S,6,S*8-8,220+RND*1200,0,-1-RND*15,RND*14+2: NEXT : RETURN
3600 COLOUR 3,2+RND*14: LINE X-8,Y,X+8,Y: RETURN
3610 LET G=.5: LET T=.5: LET M=3: LET P=1: LET N=110: RETURN
3620 FOR S=3 TO 32: SPRITE S,6,S*8-8,220+RND*500,0,-1-RND*7,RND*2+10: NEXT : RETURN
3700 COLOUR 3,2+RND*14: LET Q=RND*4+12: ANGLE 0: PLOT X-9,Y-9: DRAW Q: ANGLE PI*.666: DRAW Q: PHI PI*.666: DRAW Q: RETURN
3710 LET G=1.5: LET T=3: LET M=3: LET P=2: LET N=55: RETURN
3720 FOR S=3 TO 32: SPRITE S,0,S*8-16,200*MOD(S,6)+400+S,0,-16,2+RND*2: NEXT : RETURN
```

MEMOPAD

```
4999 REM ** Controls Screens,Etc. **
5000 CTLSPR 0,1: CTLSPR 1,1: CTLSPR 2,32: CTLSPR 3,32: CTLSPR 5,32: CTLSPR 6,2
5010 ON L GOSUB 3110,3210,3310,3410,3510,3610,3710,3810,3910
5110 VS 4: PAPER 0: INK 15: COLOUR 4,0: CLS : PRINT CHR$(29);CHR$(31);
5120 GOSUB 3000: IF Z<>-1 THEN  LET X=30+RND*200: LET Y=165 ELSE  RETURN
5130 PAPER 7: INK 12: CSR 1,0: PRINT " FUEL:      ";: PAPER 8: INK 14: PRINT " SCORE:      ";: PAPER 13: INK 14: PRINT " SH:  ";
5140 CSR 30,0: PRINT RIGHT$(STR$(SH),1);: PAPER 8: INK 14: CSR 20,0: PRINT MID$(STR$(SC),2,5);: PAPER 7: INK 12: CSR 7,0: LET F=INT(C/(Z+NS-LE)): LET C=C-F
5150 PRINT F: GOSUB 5300: RETURN
5299 REM ** Pokes M.Code Variables **
5300 POKE 64331,MOD(F,256): POKE 64332,INT(F/256)
5310 POKE 64336,M: POKE 64337,P
5320 POKE 64338,G*8: POKE 64339,T*8
5330 POKE 64333,X: POKE 64334,MOD(Y*8,256): POKE 64335,INT(Y/32)
5340 RETURN
5499 REM ** Option screen  +      **         ** Screen Order Controller **
5500 CLEAR : LET NS=7: DIM S(16,2): GOSUB 8000: LET HS=0
5510 FOR S=1 TO 16: LET S(S,1)=12*COS(S*PI/8+PI/16): LET S(S,2)=16*SIN(S*PI/8+PI/16): NEXT
5520 POKE 65535,0: POKE 64862,0: POKE 64145,132: SOUND 0,0,0: SOUND 1,0,0: SOUND 2,0,0: SOUND 3,0,0
5521 VS 5: PAPER 9: CLS : INK 4: CSR 4,23: PRINT "PRESS <F2> for instructions";
5522 CSR 4,0: PRINT "SPACE MISSION by Richard Thomas": CSR 4,1: PRINT "**************   ==============": CSR 1,3: PRINT "( You may use the numeric keypad )";
5523 LET SH=3: LET SC=0: CSR 4,11: PRINT "Starting Phase (A to ";CHR$(64+NS);") ? ";
5525 VS 5: FOR K=0 TO 600: LET A$=INKEY$: IF (A$<"A" OR A$>CHR$(64+NS)) AND A$<>CHR$(129) THEN  NEXT : GOSUB 6000: GOTO 6160
5527 IF A$=CHR$(129) THEN  GOSUB 7000: GOTO 5520
5530 LET C=10000: LET Z=ASC(A$)-65: LET K=0: PRINT A$;CHR$(7);
5533 CSR 6,13: PRINT "Speed (1 to 9) ? ";
5535 LET A$=INKEY$: IF A$<"1" OR A$>"9" THEN  GOTO 5535 ELSE  PRINT A$;CHR$(7);: POKE 64841,(9-VAL(A$))*30+1
5540 FOR LE=Z TO 9E9 STEP 0: LET LE=LE+K: IF LE<Z+NS THEN  LET L=INT(.5+MOD(LE,NS)): LET K=0: GOSUB 5000 ELSE  GOTO 5550
5545 IF L>2 THEN  ON L-3 GOSUB 3420,3520,3620,3720
5546 GOSUB 1000: GOTO 500
5550 COLOUR 0,8: COLOUR 1,1: CSR 0,0: PRINT " MISSION COMPLETE - Well Done ! ": FOR A=2 TO 32: SPRITE A,5,0,0,0,0,0: NEXT
5560 FOR A=183 TO -.1 STEP -1: COLOUR 3,RND*14+2: LINE 0,A,255,A: SOUND 2,1+A*5,10: COLOUR 4,RND*15: NEXT : LET SC=SC+5000: GOTO 6100
5999 REM ** Demo Mode **
6000 LET Z=-1: FOR L=0 TO 6: POKE 64862,9: SOUND 0,1000,700,RND*2000,0,65535,1: SOUND 1,100,70,-RND*2000,0,65535,1
6005 GOSUB 5000: IF L>2 THEN  ON L-3 GOSUB 3420,3520,3620,3720
6010 COLOUR 0,8: COLOUR 1,1: CSR 0,0: PRINT " DEMO MODE-Press any key to stop";: REM      COLOUR  4,0
6020 FOR K=0 TO 3000: IF INKEY$="" THEN  NEXT : SBUF 2: NEXT
6030 SBUF 2: SOUND 0,0,0: SOUND 1,0,0: POKE 64862,0: RETURN
6099 REM ** Print Final/High score **
6100 VS 5: PAPER 15: INK 13: CLS
6110 CSR 9,5: PRINT "High Score :";HS: CSR 9,10: PRINT "Your Score :";SC
6120 IF SC<=HS THEN  GOTO 6130 ELSE  CSR 9,15: PRINT "NEW HIGH SCORE !": LET HS=SC
6125 FOR S=1 TO 5: FOR Q=1 TO 10: SOUND 0,S*Q*10,3: SOUND 1,S*Q*10+200,S*3: SOUND 2,S*Q*10+400,S*3: NEXT : NEXT : SOUND 0,0,0: SOUND 1,0,0: SOUND 2,0,0
6130 CSR 9,23: PRINT "( Press any key )";
6140 IF INKEY$<>"" THEN  GOTO 6140
6150 IF INKEY$="" THEN  GOTO 6150
6160 FOR S=1 TO 100: NEXT
6170 IF INKEY$<>"" THEN  GOTO 6160 ELSE  GOTO 5520
6999 REM ** Instructions **
7000 VS 5: PAPER 7: INK 4: CLS
7010 PRINT "   Steer  your  spaceship  around  theobstacles using CURSOR LEFT & RIGHT andHOME (or FIRE on the right joystick) tothrust."
7020 PRINT "  To complete each screen you must landon the  platform in  the centre  of thescreen, then land on the landing pad atthe  bottom  of  the  screen
;
7030 PRINT ",  withouttouching  anything else.  If you cannoteasily reach the central platform,  youmay land  on the  landing pad,  but you";
7040 PRINT "will have to attempt the screen again."
7050 PRINT "  To  complete  the  mission  you  mustcomplete all 7 screens.You have 3 SHIPSand  10000 units of FUEL  for the wholemission.";
7060 PRINT "  Audio warning is given of lowfuel.  Your score is the amount of fuelunused  on completion  of  each screen,plus  a  BONUS  if  you  complete";
7070 PRINT "  yourmission.": PRINT "  There is a  DEMO mode if the computeris left on the option screen."
7080 CSR 3,23: PRINT "* PRESS <F3> for next screen *";
7090 IF INKEY$<>CHR$(130) THEN  GOTO 7090
7100 CLS : CSR 8,0: PRINT "SUMMARY OF KEYS": CSR 8,1: PRINT "**************"
7110 CSR 8,4: PRINT "<- : LEFT"
```

MEMOPAD

```
7120 CSR 8,6: PRINT "-) : RIGHT"
7130 CSR 1,8: PRINT "FIRE/HOME : THRUST"
7140 CSR 5,12: PRINT "SPACE : HOLD"
7150 CSR 5,14: PRINT "SHIFT : RELEASE HOLD"
7160 CSR 7,16: PRINT "ESC : (AFTER HOLD) ABORT"
7180 CSR 2,23: PRINT "‡ PRESS (F1) for option screen ‡";
7200 IF INKEY$=CHR$(128) THEN  RETURN  ELSE  GOTO 7200
7999 REM ‡‡ Define Graphics ‡‡
8000 GENPAT 1,130,16,24,28,24,16,16,16,56
8010 GENPAT 1,154,255,129,66,36,24,36,66,129
8020 GENPAT 2,154,208,208,208,208,80,80,80,34
8030 CTLSPR 6,2: FOR A=0 TO 6: FOR S=4 TO 7: GENPAT S,A,0,0,0,0,0,0,0,0: NEXT : NEXT : GENPAT 4,0,0,0,0,24,36,126,219,126
8040 GENPAT 4,1,128,65,5,24,80,138,34,33
8050 GENPAT 5,2,0,0,108,36,56,24,64,0: GENPAT 5,3,0,0,44,50,80,0,0,0
8060 GENPAT 4,4,27,60,126,255,255,126,60,25: GENPAT 6,4,0,96,136,18,105,22,32,128
8070 GENPAT 4,6,28,62,127,65,65,34,34,20: GENPAT 5,6,8,28,62,107,62,20,34,20
8080 SBUF 2: SOUND 0,0,0: SOUND 1,0,0: SOUND 2,0,0: SOUND 3,0,0
8090 PRINT CHR$(27);"U";CHR$(255);: RETURN
65534 REM ‡‡ Auto-SAVE routine ‡‡
65535 CLEAR : SBUF 2: SAVE "SPACE MISSION": RUN
```

Sound & Vision

THIS NEAT LITTLE PROGRAM DEMONSTRATES WHAT IS POSSIBLE WHEN UTILISING THE SOUND AND VISION CAPABILITIES OF YOUR MEMOTECH IN UNISON WITH THE LIMITLESS BOUNDS OF THE HUMAN IMAGINATION. LET'S SEE WHAT OTHER MEMBERS CAN COME UP WITH!

```
0 REM ******************************
1 REM ***** 'Sound & Vision'  ****
2 REM ***** by Richard Thomas ****
3 REM ******************************
10 DIM A$(3,29)
20 LET A$(1)="@EGHLJEEQQQOLLQQGHLJEEEEEDDE"
30 LET A$(2)="@99<<>>AA99<<@Q99<<>>AA99@@9"
40 LET A$(3)="@<<@@AAEE<<@@DQ<<@@AAAA<<DD<"
100 FOR S=5 TO 13 STEP 2: SBUF 57
110 POKE 64862,0: FOR C=0 TO 3: SOUND C,1,0,0,0,10,1: NEXT : POKE 64862,15
120 REM ***   SOUND  one-liner ***
130 FOR N=1 TO 28: FOR C=0 TO 2: SOUND C,8192/2^((ASC(A$(C+1,N))-57)/12),0,0,242,4,1:
    SOUND C,0, 0,0,-8.46-25*MOD(N,2)-75*(MOD(N,14)=0),0: NEXT : NEXT

200 REM ***  GRAPHIC one-liner ***
210 VS 4: COLOUR 4,8: PAPER 1: CLS : PLOT 130,24: ANGLE 0: FOR I=2 TO 15: INK I: PHI PI/7: DRAW
33: FOR A=1 TO S: PHI PI+PI/S: DRAW 33: NEXT : NEXT
220 CSR 25,20: PRINT "<RET>": CSR 26,21: PRINT "when": CSR 25,22: INPUT "ready";B$: NEXT
```

MEMOPAD

HIGH SCORES

ASTRO PAC	213,430	Michael Hunt
KNUCKLES	1,147,360	T. Erikson
CHAMBERIODS	Comp. 4 mins	T. Erikson
MISSION ALPHATRON	175,340	Matthew Moss
TAPEWORM	175,980	Richard Franks
TOADO	356,414	John Quinn
POT HOLE PETE	106,630	Richard Franks
MAXIMA	1,479,710	S. Olauder
STAR COMMAND	140,430	Ian Nichols
OBLOIDS	62,400	M. Hurley
PHAID	26,000	Sally Street
KILOPEDE	82,253	Richard Nash
3D TACHYON FIGHTER	12,500	C. Walker
CONTINENTAL RAIDERS	106,240	Sean Haverty
BLOBBO	148,283	E. Mahon
QUANTUM	14	Ian Cartwright
QOGO 2	205,000	R. Siddall
MINEFIELD	2,100	C. Walker
FLUMMOX	251,510	C. Walker
TURBO	18,610	Michael Hunt
FATHOMS DEEP	3,450	Matthew Moss
AGROVATOR	675,000	P. Howard
FIREHOUSE FREDDIE	29,620	T. Erikson
QOGO	43,960	T. Erikson
ARCADIANS	25,900	Adrian Johnson
MISSILE COMMAND	27,580	Adrian Johnson
LITTLE DEVILS	34,320	Leslie Banks
FELIX IN THE FACTORY	14,740	Peter Crighton
HUNCHY	8,457	John Quin
SON OF PETE	17,233	T. Erikson
HAWKWARS	25,800	Gordon Hurd
ESCAPE FROM ZARCOS	76 Items	G. Hill
SALTY SAM	40,642	Andrew Johnson
MISSION OMEGA	10,850	A. Knott & S. Paine
ICEBURG	17,431	Alan Dobson
SNOWBALL	1,000	Victor Stepney
EMERALD ISLE	768/1000	Victor Stepney
SUPERBIKE	23.9kms	A. Clark
ROLLA BEARING	27,000	Victor Stepney
DR. FRANKIE	65,435	J. Graham
TARGET ZONE	17,470	D.J. Chamberlain
MINER DICK	22,520	R. Siddall
JUMPING JACK	26,120	A. Miller
SURFACE SCANNER	72,060	T. Erikson
CAVES OF ORB	496/500	V. Stepney
SEPULCRI SCALERATI	8,000	Andrew Miller
SMG	105,400	Clare Townsend
RETURN TO EDEN	1,000	Andy Crick
QUAZZIA	26,660	Andrew Miller
OBLITERATION ZONE	32,670	Alan Dobson
ASTORMILLION	142,342	D.J. Chamberlain
CRYSTAL	32,425	Gordon Hurd (COMP)
DRIVE THE CEE?5	12,907	V. Stepney
HIGHWAY ENCOUNTER	123,120	
KARATE KING	4,570	G. Hill
DOWNSTREAM DANGER	8,976	G. Hill
DOODLEBUGS	4,340	A. Miller
THE WALL	53,500	A. Miller
COMBAT	47,690	A. Miller

MEMOPAD

PRICE LIST

HARDWARE

DESCRIPTION	MEMBERS PRICE	NON MEMBERS PRICE	CARRIAGE
COMPLETE CP/M PACKAGE			
1 X 1 MBYTE 3.5" INDUSTRY STANDARD DISC DRIVE, 500K FAST ACCESS RAM DISC CP/M 2.2 OPERATING SYSTEM. 256K RAM. 12" GREEN SCREEN MONITOR CENTRONICS STANDARD PRINTER I/F POSITIVE ACTION KEYBOARD. COLOUR MONITOR OUTPUT. TWO JOYSTICK I/F.	359.95	399.95	20.00
PURCHASES INDIVIDUALLY (basic system)			
256K COMPUTER PLUS TAPE OPERATING SYSTEM	89.95	99.95	10.00
CP/M SYSTEM			
1 X 1 MBYTE 3.5" DRIVE + 512 SILICON DISC + 80 COL + CP/M + N.W.	237.59	264.00	10.00
HX 12" GREEN SCREEN MONITOR	85.49	95.00	10.00
TWIN RS232 INTERFACE (UPGRADE)	26.96	29.95	3.00
FDX 2 X 1 MBYTE CP/M + 2 MBYTE SILICON DISC.	877.50	975.00	10.00
32K MEMORY EXPANSION	37.95	39.95	3.00
64K MEMORY EXPANSION	47.45	49.95	3.00
128K MEMORY EXPANSION	75.95	79.95	3.00
NEWWORD ON ROM	37.95	39.95	3.00
PASCAL ON ROM	37.95	39.95	3.00
RS232 INTERFACE (FULL BOARD)	37.95	39.95	3.00

MEMOPAD

SIDISC PRICES

1 X 1 MBYTE	161.10	179.00	10.00
1 X 2 MBYTE	304.20	338.00	10.00
1 X 3 MBYTE	542.40	636.00	10.00
PRINTER CABLE	11.65	12.95	0.50

SPECIAL NOTES

Silicon Discs can be factory fitted for an extra £30.00 (U.K. Only).
FDX Twin Systems require RS232 Comms Board.
Carriage is applicable to U.K. orders only.
Always quote the type of computer owned Eg: MTX 500, 512 or series 2 when ordering hardware.

** PLEASE NOTE ALL PRODUCTS WHICH ARE NOT MENTIONED ON THIS LIST ARE NO LONGER AVAILABLE **

THE ISSUE OF THIS PRICE LIST CANCELS ALL PREVIOUS OFFERS

MANUALS

CRIB CARDS	£1.50	ROM CALLS INFO SHEET	0.50
RST10 CALLS INFO SHEET	0.50	INTERRUPTS INFO SHEET	0.50
MTX SERVICE MANUAL	£9.95	MTX NEW USER MANUAL	£8.95
V.D.P. MANUAL	£7.95	D.D.T. MANUAL	£2.50

DISC BASED LISTINGS

MTX ROM LISTING	5.25" -	£15.95	3.5" -	£21.95
SDX DISC CONTROLLER	5.25" -	£9.95	3.5" -	£12.95

ADVERTISING IN THE MEMOPAD

SMALL ADVERT.	£5.00	1/8 PAGE	£27.50
1/4 PAGE	£45.00	1/2 PAGE	£80.00

ALL CHEQUES SHOULD BE MADE PAYABLE TO ORION SOFTWARE

****

SUNDRIES

DISC HEAD CLEANER	3.5" -	£18.25	5.25" -	£16.95
HOME COMPUTER MAINTAINANCE KIT	-	£21.50 + 50p P+P		
SMALL DISC BOX (HOLDS 40)	-	£16.00 + 75p P+P		
LARGE DISC BOX (HOLDS 80)	-	£19.95 + 75p P+P		
DMX 80 PRINTER RIBBONS	-	£8.96		

MEMOPAD

SOFTWARE

U = Utility E = Educational L = Language
G = Game J = Joystick Compatible B = Business

3D TACHYON FIGHTER	G+J	ANY	5.95
AGROVATOR	G	ANY	5.95
ALICE	G	ANY	5.95
ASTROMILON	G+J	ANY	5.95
ASTROPAC	G+J	ANY	5.95
KILLER TOMATOES	G	512	7.95
BLOBBO	G+J	ANY	5.95
BOUNCING BILL	G+J	ANY	5.95
BRIDGE	G	ANY	5.95
CAVES OF ORB	G	512	5.95
CHAMBEROIDS	G+J	512	6.95
CHESS	G	512	8.95
COMBAT	G	512	3.95
CRIBBAGE	G	512	5.95
CRIBBAGE (SDX)	G	512	9.95
CRYSTAL	G	512	6.95
DISASM	U	512	7.95
DOODLEBUG	G+J	ANY	5.95
DOWNSTREAM DANGER	G+J	512	6.95
DR. FRANKIE	G+J	ANY	5.95
DRIVE THE CEE 5	G+J	ANY	6.95
EDASM	U	512	7.95
EDASM SDX (DISC)	U	512	8.95
EMERALD ISLE	G	ANY	6.95
ESCAPE FROM ZARCOS	G+J	512	6.95
EXTENDED BASIC	U	ANY	7.45
F1 SIMULATOR	G+J	512	4.95
FATHOMS DEEP	G+J	512	6.95
FIG FORTH	L	512	15.75
FIG FORTH SDX (DISC)	L	512	15.75
FIREHOUSE FREDDIE	G+J	512	5.95
FIRST LETTERS 1	E	512	5.95
FLUMMOX	G+J	512	6.95
FOOTBALL MANAGER	G	ANY	6.95
FOOTBALL MAN. 3.5"	G	512	9.95
FOOTBAL MAN. 5.25"	G	512	8.25
GHOSTLY CASTLE	G	ANY	3.45
GOLDMINE	G+J	ANY	5.95
GRAPHICS	U	ANY	5.95
HEI-MATHS	E	512	7.95
HIGHWAY ENCOUNTER	G+J	512	7.95
HUNCHY	G+J	ANY	5.95
ICEBERG	G+J	ANY	5.95
JUMPING JACK FLASH	G+J	512	5.95
KARATE KING	G+J	ANY	6.95
KILOPEDE	G+J	ANY	5.95
KNUCKLES	G+J	ANY	5.95
LITTLE DEVILS	G+J	ANY	5.95
MISSILE COMMAND	G+J	ANY	5.95
MATHS 1	E	512	8.95
MAXIMA	G+J	ANY	6.95
MEMOCHEQUE	U	512	6.95
MEMOSKETCH	U+J	ANY	7.95
MEMOSKETCH SDX	U+J	512	8.95
MISSION ALPHA.	G+J	ANY	5.95
MISSION OMEGA	G+J	ANY	5.95
NEMO	G+J	ANY	6.95
OBLIT. ZONE	G+J	ANY	6.95
OBLOIDS	G+J	ANY	5.95
PAINTBOX	U	512	5.95
PAYROLL	B	512	11.25
PHAID	G+J	ANY	5.95
PONTOON	G	ANY	5.95
POT HOLE PETE	G+J	ANY	5.95
PRO WORD	B	512	12.95
PURCHASE LEDGER	B	512	9.75
QOGO	G+J	ANY	5.95
QOGO 2	G+J	512	6.95
QUANTUM	G+J	ANY	5.95
QUAZZIA	G+J	512	6.95
QUEST 1	G	ANY	5.20
REVERSI	G	512	5.95
ROLLA BEARING	G+J	512	6.95
RUTHLESS B.	G	512	3.45
SALES LEDGER	B	512	10.75
SALTY SAM	G+J	ANY	5.95
SELF TEST	U	ANY	7.95
SELF TEST (SDX)	U	ANY	9.95
SEPULCRI	G	512	6.95
SMG	G+J	512	6.95
SNAPPO	G+J	ANY	5.95
SNOOKER	G	512	5.95
SON OF PETE	G+J	ANY	6.95
SUPA CODER	U	512	7.95
SUPER BIKE	G+J	ANY	5.95
SUPER MINEFIELD	G	ANY	5.95
SURFACE SCANNER	G+J	512	6.95
TAPE TO DISC	U	512	6.95
TAPEWORM	G+J	ANY	5.95
TARGET ZONE	G+J	512	6.95
THE WALL	G+J	512	5.95
TOADO	G+J	ANY	5.95
TURBO	G+J	ANY	5.95
USER BASIC	U	512	8.96
USER BASIC (SDX)	U	512	9.95
UTILITIES (SDX)	U	512	9.95
VERNON & VAMPS.	G	ANY	5.95
WORD AND PICTURE	E	512	5.95

This file was downloaded from

www.primrosebank.net

Abridged Terms & Conditions (Downloads)

Disclaimer

www.primrosebank.net, (***the website***) is provided by Dave Stevenson as a service to the public, is provided "as is" and carries no warranties, expressed or implied, of any kind.

Dave Stevenson is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within the website or made available for download. Whilst the information contained within the website site is periodically updated, no guarantee is given that the information provided on the website is correct, complete, and up-to-date.

A number of articles on the website contain technical data and practical guidance which may be of use in testing and maintaining various items of vintage computer and electronics hardware. Such articles are not intended to cover all aspects of the tasks involved and may omit essential information, including necessary safety precautions. Performance of the tasks described may risk damage to equipment and/or people. The reader is responsible for ensuring that he/she is capable of performing the tasks described and well as assessing the inherent risks involved and taking appropriate measures to mitigate such risks.

Dave Stevenson expressly disclaims all liability for, damages to equipment or injury of any kind arising out of use of such technical data and guidance.

Unless otherwise noted, all data on the website is deemed to be ***Copyright (c) Dave Stevenson, 2009-2013***

You are hereby granted permission to download data and software from the website ***for your own personal use***. ***Redistribution of any content from the website without written authorisation from Dave Stevenson is expressly forbidden. You are also expressly forbidden from offering for sale any material obtained from the website.***

As far as possible, information included on the website from other sources has been credited to the respective author and/or publisher. The majority of content on the website is derived from material first published in the 1980s. ***This material is likely still under copyright of the original author and/or publishers.*** The authors and/or publishers may not have given express permission to copy, transmit or make this information available for download, but I believe that they would have no objection to this archive information being placed into the public domain.

However, should the author and/or publisher of the original material find any content on the website for which they wish to assert their rights, they should notify Dave Stevenson (by e-mail to: webmaster@primrosebank.net) who would be pleased to enter into a dialogue to agree a satisfactory resolution of their concerns.

If you obtained this file as part of a paid-for package, you have been scammed! I suggest that you request a refund from the seller, please also advise Dave stevenson at the e-mail address above.

PLEASE NOTE It is not always possible to put tape based software on disc, please check with our sales staff before ordering any of the above on 5.25" or 3.5". If the software will transfer a charge of £1.50 for 5.25" and £3.00 for 3.5" will be added to the tape price.

DISC BASED SOFTWARE

MEMOSKETCH	3.5"	(CPM)	£9.95	5.25" (CPM)	£7.95
ROLLA BEARING	3.5"		£9.95	5.25"	£8.99
FRANTIC FREDDIE	3.5"		£9.95	5.25"	£8.99
SOURCE DISC	3.5"		£9.95	5.25"	£9.95
CONTACT III	3.5"		£14.95	5.25"	£14.95
FAB FIVE	3.5"	(CPM)	£12.95	5.25" (CPM)	£7.99
DISC ONE	3.5"	(CPM)	£11.95	5.25" (CPM)	£7.99
DISC TWO	3.5"	(CPM)	£11.95	5.25" (CPM)	£7.99
DISC THREE	3.5"	(CPM)	£10.95	5.25" (CPM)	£7.50
DISC FOUR	3.5"	(CPM)	£9.50	5.25" (CPM)	£7.50

DISC ONE

KNUCKLES
OBLOIDS
QUASAR
TOADO
TURBO

DISC TWO

3D TACHYON FIGHTER
MISSION ALPHATRON
ASTROPAC
NEMO

DISC THREE

STAR COMMANDER
PACMAN
MINEFIELD
POTHOLE PETE

DISC FOUR

REVERSI
CHESS

FAB FIVE

3D TACHYON FIGHTER
MISSILE COMMAND
TIME BANDITS
BACKGAMMON
ARCAZIONS

Please make all cheques payable to Orion Software

ATTENTION ALL NEWWORD USERS .....
WE HAVE A SMALL QUANTITY OF BOOKLETS THAT DESCRIBE ALL YOU NEED TO KNOW ABOUT CUSTOMISING NEWWORD FOR YOUR OWN NEEDS.
THIS BOOKLET LISTS ALL *THE USER AREA PATCHES* AND WHAT IS EXPECTED FROM THE OPERATOR TO INCLUDE CUSTOMISED COMMANDS
WHILE STOCKS ARE AVAILABLE 3.00p

CP/M - 80 SYSTEMS ONLY

OTHER BOOKS ON NEWWORD WILL BE AVAILABLE SHORTLY .....

PICK N' MIX

HERE'S YOUR CHANCE .... TAKE YOUR PICK
ANY FIVE PROGRAMS
ONLY 10.00

ASTROPAC : BLOBBO : BACKGAMMON : NEMO
MISSION ALPHATRON : STAR COMMAND : OBLOIDS
REVERSI (512 only) : QOGO : ASTROMILLION

PLEASE ADD 50p P&P EUROPE ADD 1.25p
OTHERS PLEASE ADD 3.50p

THE NEW PANASONIC KXP1082 MATRIX PRINTER

NOW AVAILABLE FOR USE WITH YOUR M.C.L COMPUTER. THIS PRINTER WILL OPERATE AT AN INCREDIBLE 160 CHARACTERS PER SECOND !
ALSO HAS A NEAR LETTER QUALITY MODE.

+ Operator accessible print mode selector 4 position [std/pgm Courier, Bold, PS, Comp]

+Near Letter Quality in all modes and pitches

+High resolution dot-addressable graphics compatible with most printer software.

+Snap-in ink ribbon that will print up to 3 million characters. Also available in RED BROWN BLUE

+Bit images [Dots per inch] 60-72-80-90-120-144-240.

NOW AVAILABLE 216.00 + 10.00 carriage.

Barclaycard & Visa welcome ring *Barbara* on 0282/831695

ORION SOFTWARE The Northbridge Centre, Elm Street, Burnley BB10 1PD

** STOP PRESS NEWS **

IN VIEW OF THE POSSIBLE POSTAL STRIKE
CUSTOMERS ARE ADVISED TO PLACE THEIR
X-MAS ORDER NOW.
