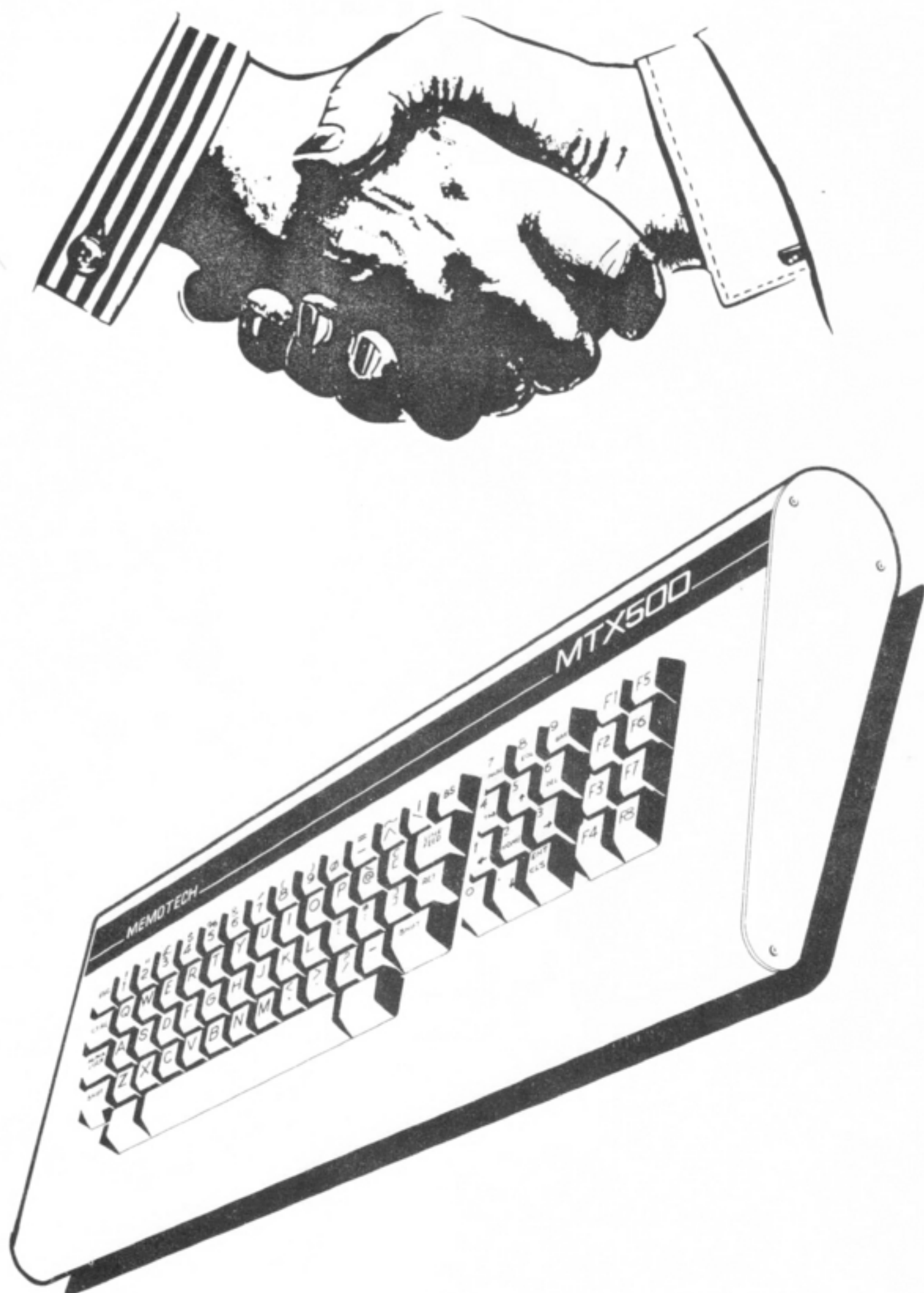


memorad

Memotech Computer User Club Magazine

ISSUE NO. 6



MEMOTECH
MTX
SERIES

EDITORIAL

For those members who are a little uncertain about the future of Memotech I would like to put the record straight. The Company is doing very nicely - in spite of itself - and are going ahead with various projects that will reach you later in the year. I am one of the first to criticise Memotech for their apparent inability to read the home market, however, I must agree with their present policy of non-advertising. I say this with two points in mind:

1. The recent demise of Oric and the uncertain future of Acorn - both companies have stumbled in to deep water by failing to keep their overheads in check.

2. The lack of support by the computer magazines. How can any magazine justify running two programs for a machine that is no longer available, and yet, totally ignore a very popular computer? This happened in one magazine this week! Even when we do get a mention the material used is completely dated, proving that the magazines are completely out of touch. If you don't believe me take a look at this month's **What Micro**. The software reviewed in that magazine are some of the very first titles released. No mention of POTHOLE PETE, AGROVATOR, and the like!

The Company's stock reply to advertising departments is, 'when the magazines start to recognise the MTX, the Company will advertise within their pages.' - A fair enough comment. It is about time we stopped moaning about the lack of support and started to do something about it. I urge you all to spend 10 mins - not long is it - and reiterate the Company's policy by stating that you will no longer take delivery of XX magazine until you find that your computer is supported. If every member of Genpat wrote to two magazines in this way, they would soon take notice! **SO PLEASE! DON'T JUST READ IT DO IT AND DO IT NOW!**

GENPAT'S HIT LIST

- 1 YOUR COMPUTERworst offender.
- 2 H.C.W
- 3 PRACTICAL COMPUTER PATHETIC!
- 4 WHAT MICRO
- 5 P.C.N supported MTX from day one now no coverage.
- 6 PERSONAL COMPUTER NEWS
- 7 WHICH MICRO at least they have tried!

Because pathetic is still on my mind, I'll approach another subject. Just what do most of you do with your computers? Look through the pages of Memopad, you will find the same names cropping up again and again. The lack of participation by members has us scratting around trying to find interesting programs for you. So after you have written to the magazines would you be kind enough to get that program, you've been hiding, out of the cupboard and send it to us at the club? Thanks. Although I'm having a go, I still think that you're the best set of computer users I have ever been associated with - it's nice to have a few thousand friends!

Incidentally, before I close, watch out for **FLUMMOX**. I kid you not, this is one of the best arcade style games I've seen in a long, long while.

HARDWARE

Floppy 5 1/4" Discs	- Verified to 96 TPI (for 10 in plastic library case)....#17.50 ★
Microvitec 1431/AP/MS	- PAL/TTL with audio.....#225.00
Microvitec 1431/MS	- TTL.....#205.00
Cosmos 80	- High density 80 cps matrix printer. Friction & tractor feed. Dot hi-res graphics...#199.00
Ensign	- Near letter quality printer 165 cps. Bi-directional.....#335.00
Seikosha GP550A	- Matrix printer.....#205.00 ★
Silver Reed	- High quality daisy wheel....#699.00
Centronics cable	- For the above.....#12.95
Printer ribbons	- For the DMX80.....#8.50
Dust cover	- Keep your computer clean....#3.50

Competition

ONE LINER

```

5 REM PHILIP DANIELLS
10 REM You play the part of an adder. The idea is to guide your way through
15 REM the field of poisonous toadstools and to score as many points as you can
20 REM You move as follows :-
25 REM Z - moves you left
30 REM X - moves you right
35 REM Enter the program using the abbreviated form of the keywords
40 REM
100 LET X=10: LET S=0
110 CSR RND$40,23: PRINT X: LET S=S+1: LET A$=INKEY$: LET X=X+(A$="Z")-(A$="X"): CSR X,10: PRINT " $ ": CSR X+(A$="Z")-(A$="X")+1,11:
0 ELSE PAUSE 5000: CLS : PRINT "SCORE=";S
IF SPK$=" " THEN GOTO 11

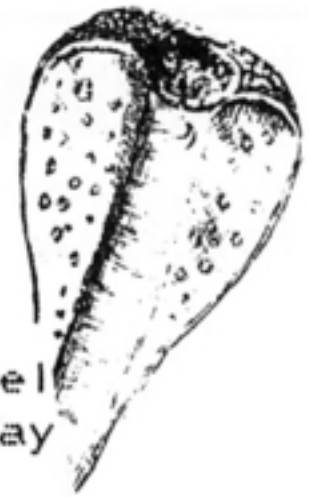
10 REM << PROGRAM TO DRAW A CHEQUERED BOARD >>
20 REM << AS USED IN CHESS AND DRAUGHTS >>
30 REM by JOHN GRAYSON
100 GENPAT 0,33,255,255,255,255,255,255,255,255: CRVS 2,1,8,4,16,16,32: VS 4: PAPER 3: CLS : VS 2: PAPER 15: CLS
110 REM
120 REM +++++START OF PROGRAM+++++
130 REM
140 VS 2: INK 1: FOR V=2 TO 14 STEP 4: FOR H=0 TO 12 STEP 4: FOR J=0 TO 1: CSR H,V+J: PRINT "!!";: CSR H+2,V+J-2: PRINT "!!";: NEXT : NEXT : NEXT
150 GOTO 150

10 REM QUADRATICS by SIMON WRIGHT
20 GENPAT 1,129,0,0,0,24,4,8,16,28
30 INPUT A,B,C: CLS : LET D=SQR(B*B-4*A*C): PRINT " ";CHR$(129): PRINT A;"x";CHR$(44-SGN(B));ABS(B);"x";CHR$(44-SGN(C));ABS(C);"=",(-B+D)/A/2;
" or";(-B-D)/A/ 2

```


REVIEW

Lords Of Time



One only has to know that 'Lords of Time' comes from Level 9, and you are sure of a superb adventure - some might say this is their all round best.

The game requires you to travel through 9 time zones - including Modern, Prehistoric, Roman, Medieval and Futuristic Times. In each zone you collect a special object - you are told these in the accompanying booklet to the game. In the last time zone you must then use the 9 objects to defeat the Evil Time Lords who have been meddling with Earth's history.

As with the other Level 9 adventures (at least for the MTX) it is a text adventure, but what text! All locations have such imaginative descriptions that you can really imagine yourself being there. When it doesn't understand something you've entered, the replies are both appropriate and varied.

All in all a superb game, which will keep you occupied for hours if not weeks! Definitely a must if you like adventure games.



Competition

One-Liners

This is a one-liner from Gary Wilkinson

```
10 LET G$="": CLS: INPUT "ENTER A WORD ";A$: LET X=LEN (A$): FOR T=X TO 1
STEP -1: LET B$=MID$(A$,T,1): LET G$=G$+B$: NEXT T: PRINT "REVERSED WORD
IS ";G$
```

A couple of one-liners from Eric Roy

[Decimal to Binary - 0 to 255]

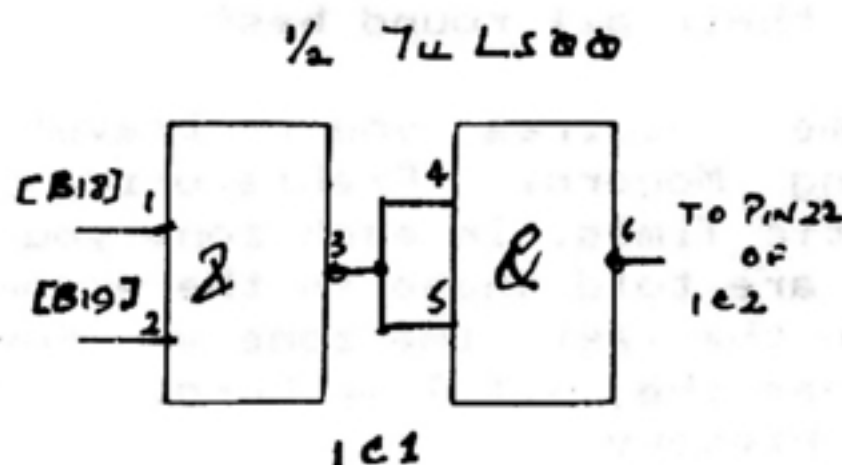
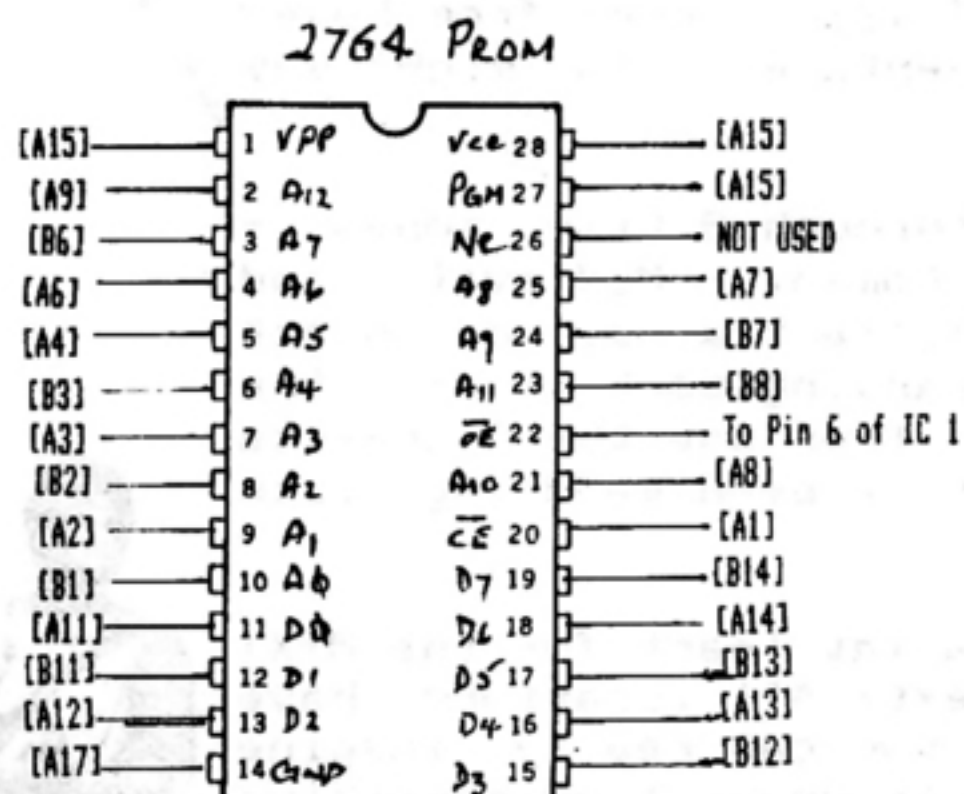
```
10 CLS: INPUT "Decimal No ";D: LET B$="": FOR R=9 TO 1 STEP -1: LET B=D-
INT(D/2)*2: LET B$(R)=CHR$(B+48): LET D=INT(D/2): NEXT: PRINT B$(2,9)
```

[Binary to Decimal]

```
10 CLS: INPUT "Binary No ";B$: LET D=0: LET C=-1: LET N=LEN (B$)+1: FOR
R=1 TO 2 STEP 0: LET C=C+1: LET N=N-1: IF N<1 THEN PRINT D ELSE IF
B$(N)="0" THEN NEXT ELSE LET D=D+2^C: NEXT
```


HARDWARE CARTRIDGE PORT

L.R. WHALLEY



*** NOTE Figures enclosed in [] refer to edge connector as shown in manual (Page 208)

The 3 ways of running the PROM :-

Method 1

Start your code at the location 200C hex (8204 dec)
To run your program type (in BASIC) ROM 7 <RET>

Method 2

By inserting 0BH to 01H in the first 8 locations (2000H to 2007H)
Start your code at 2010H
This will auto run on switch on

Method 3

Start your code at 2000H (8192 dec), or any other location you want between 2000H and 3FFFH.
To run your code you must select ROM 7 (cart. port) with a short machine code program, as per my example, you must also reset to original rom setting before returning to basic, to avoid crashing the system.

100 CODE

```

SR0M7:  LD A, (#FAD2)      ; get ROM select
        PUSH AF           ; save original
        OR #70            ; set ROM 7
        LD (#FAD2), A
        OUT (#00), A      ; enable ROM 7 (cart. port)
START:  Your code
        containing calls
        to ROM 7
        >>>>>>>>
        >>>>>>>>
RR0M7:  POP AF            ; get original ROM settings
        LD (#FAD2), A     ; reset
        OUT (#00), A
        RET               ; return with original ROM setting
    
```

A simple illustration of my use of the cartridge slot:

I have to date 45 routines housed in my PROM including and index to all routines and their respective addresses.

100 CODE

```
BEGIN:      JR START          ; goto program start
TEMPA:      DS #01            ; temp store for original rom setting
SR0M7:      LD A, (#FAD2)      ; get rom setting
            LD (TEMPA), A      ; save it
            OR #70             ; set ROM 7
            LD (#FAD2), A      ; store it
            OUT (#00), A       ; enable ROM 7
            RET

RR0M7:      LD A, (TEMPA)      ; get original rom setting
            LD (#FAD2), A      ; store it
            OUT (#00), A       ; reset to original rom setting
            RET

START:      CALL SR0M7         ; enable ROM 7
            CALL #2130         ; print routine (terminator=#FF)
            DB #0C, 'MY SUBROUTINES', #0A, #0D, #FF ; cls and print it
            LD C, #E1          ; #E1 - E is ink, 1 is paper
            CALL #20C0         ; fill colour table with #E1 (new colour)
            CALL #23B0         ; select VS 5 and clear it
            CALL RR0M7         ; restore original rom setting
            RET                ; return to BASIC
```

This program does nothing in particular it just illustrates my use of the cartridge port. The program enables ROM 7 (cart. port), calls the print subroutine, clears the screen and then prints "MY SUBROUTINES", prints a line feed, prints a carriage return, fills the VRAM colour table, with the contents of the 'C' register (#E1) - which changes the ink to grey and paper to black - and resets rom settings and returns to BASIC. ★



FIREHOUSE FREDDIE from OXFORD DATA

REVIEW



Firehouse Freddie is in the style of Pot Hole Pete and its ilk and if that is the type of game you like then it shouldn't be a disappointment. Full instructions are given within the program - a feature which is sadly lacking from many software packages - and these can be returned to at the end of each game. The idea behind Firehouse Freddie is to collect a key and go to the exit door, which takes you to the next screen. Between Freddie and this goal are moving ladders, conveyor belts, pieces of furniture, fruit, greasy poles, ropes, lifts, emergency fire apparatus, a cauldron of fire and some furry monsters.

The graphics are impressive with an intelligent choice of colours which came over excellently on a Ferguson TX. Points are scored when Freddie climbs up ladders, eats some fruit or jumps over the furniture. A bonus system allows you to score extra points but the number scored depends upon how quickly you complete the screen.

There's a high score table for up to eight names (max 24 characters each) and should the last score not make the top eight it is displayed as well. Someone with a modicum of skill should get passed screen 3, unfortunately I did not reach such dizzy heights of success. Firehouse Freddie is a game I would not hesitate in recommending. ★

BOUNCING BILL from SYNTAX SOFTWARE

REVIEW



"Simplicity itself" would be the phrase to sum up Bouncing Bill but this is not a criticism, far from it. Many games these days are sold on their fantastic graphics but it is the game itself which is the most important factor in keeping the interest of the user. Bouncing Bill consists of eight horizontal lines across the screen with Bill standing at the bottom. Gaps, which move up and down the screen, appear in the lines and the idea is to jump through these gaps until you get Bill to the top. Your timing has to be perfect or else you bang Bill's head - which is fatal at the bottom but just stuns him on higher lines. The stunning effect puts him out of action for a few seconds, which can be a problem if a gap appears on the line underneath him. If this happens on the top line it is quite feasible to find yourself at the bottom again very quickly. Once Bill is on the top line, jumping up completes the screen.

The second screen introduces a white disc which makes its way down the screen towards you. It causes Bill to be stunned as well, if it runs into him. Screen three has two white discs, but as for screen four onwards I didn't manage to get that far. Bouncing Bill is the sort of game any member of the family can learn how to play in a few minutes. It is great fun and gets lots of laughs when it is being played. Bouncing Bill is highly recommended. ★



PONTOON AND BLACKJACK from CONTINENTAL SOFTWARE

REVIEW



The cassette insert for this program claims that it is written in machine code when in fact it is written in BASIC. As I'm in one of my more generous moods I'll assume this is an honest mistake and not deliberate deception by Continental. The game itself comes with comprehensive on-screen instructions for both pontoon and blackjack. The program allows all the respective features of each card game (stand or stick, draw or twist, betting etc.) and it is done very effectively. I particularly like the "double down" option in Blackjack which allows you to double your bet and receive a new card. Paul Wood has done a good job with this program and it is enjoyable to play, it's just unfortunate that Continental insist on charging #6.95 for a computer-based card game. ★



Sale

CGP-115 4 COLOUR PRINTER/PLOTTER
WITH MTX COMPATABLE CABLE, PSU,
EXTRA PAPER AND PENS. SERIAL AND
PARALLEL INTERFACES, 40 or 80 COLUMN
TEXT. POTHOLE PETE FREE WITH PRINTER
ONLY 99 pound. TEL : 9-33-2824 AFER 6
p.m. (THE SOUND TABLE WAS PRINTED
WITH IT.)

Low

SPOOLER by MEMBRAIN SOFTWARE

REVIEW



Our friends at Membrain have been busy again, and have sent me an excellent utility package 'SPOOLER'. This piece of software allows you to make Hi-res screen dumps but with a difference: you can carry on computing while the printer gets on with the printing ! You can even write to the screen once you have told the printer to make a dump because Membrain have been very crafty and used a buffer to store the data. You can also make screen dumps of the Front Panel.

Once loaded, the program is simplicity itself to use. Typing 1 will dump the Panel, and typing 9 will dump the Hi-res screen. There is a mistake with the on screen instructions which say you must type 0 for a Hi-res dump. This threw me for a moment, but as my copy is a review sample I can only hope that the error has now been corrected.

An excellent program and at £3.60 to members who can not afford to have this program. Available by return of post from Membrain Software, 25, High Road, Redworth, Newton Aycliffe, Co. Durham. ★



MTX BASIC EXTENSIONS By Sentient Software

REVIEW



Once in a while a real gem surfaces and this is the case with this piece of software. Written by John Mullins this Basic Tool Kit is just what the doctor ordered. Although my copy was a pre-production version, and other commands have still to be added, I realised immediately how useful this package will be to the Basic programmer.

Some of the extensions supported are: .SSNEW_ - clear memory but preserve the extension program. .BSAVE_ Save a block of code to tape in binary format. .PAINT_ This will fill an area of the screen with the desired colour starting at the X,Y co-ordinates specified in the parameters. .DPOKE_ allows you to perform a two byte poke - no more working out the MSB LSB.

Sentient have also added commands to the Front Panel: Fill. if you press F when using the Panel you will be prompted to specify an end address and the character you wish to fill the memory with. VRAM read. This command will allow you to copy Vram into Ram. Write Ram to Vram. This is by far my favourite command and is very useful for getting data into VRAM from Basic.

The price has not been announced yet, but knowing Sentient it will not be an 'arm and a leg'.... we will keep you informed of the intended release which will be within the next few weeks. Needless to say, Genpat will have full stocks of this as soon as it is available. ★

PROGRAM LISTING

Dear Sir,

I am only a beginner in writing programs so please do not hesitate to throw these papers in the waste-paper basket if you think this is worthless for GENPAT.

Thanks to the information I received from Mr Hook about RST IO and CALL &79 I could realize this small and thumb TERMINAL EMULATION program for the MTX.

I tested it on a DEC PDP-II/34 at baudrates up to 1200 baud. At higher BR I lost characters.

```

10 CSR 9,0: PRINT "TERMINAL EMULATION"
20 CSR 13,3: PRINT "FOR MTX 512"
25 CSR 8,6: PRINT "-Type <ESC> to EXIT mc": CSR 8,7: PRINT "-Use channel RS232 - 0"
27 FOR I=0 TO 38: PRINT "*";: NEXT
30 CSR 0,12: INPUT "BAUD RATE: ";BR
40 BAUD 0,BR
45 PRINT CHR$(30): REM CURSOR ON
50 CLS
200 CODE

```

```

40D8 START: LD HL,START
40DB EI
40DC REC: IN A,(#E);CTL RS A
40DE BIT 0,A;DATA AVAILABLE?
40E0 JR Z,KEY
40E2 IN A,(#C);GET DATA RS A
40E4 OR A;STILL NO DATA
40E5 JR Z,KEY;JMP KEYB IF NOT
40E7 JR SCRN;DISPL ON SCREEN
40E9 JR REC
40EB KEY: XOR A;CLEAR A
40EC CALL #79;KEYSCAN
40EF JR Z,REC
40F1 LD E,A
40F2 CP #1B;ESC ??
40F4 JP Z,END;YES, STOP
40F7 JR OUTPUT;OUT TO RS A
40F9 OUTPUT: IN A,(#E);CTL RS A

```

```

40FB BIT 2,A;LOOP UNTILL TRANSM
40FD JR Z,OUTPUT;BUFFER IS EMPTY
40FF LD A,E;LOAD DATA IN ACC
4100 OUT (#C),A;OUT TO RS A
4102 JR REC
4104 SCRN: LD C,A
4105 LD B,0
4107 RST 10
4108 DB 192
4109 JR REC
410B END: RET
410C RET

```



```

210 CSR 10,24: INPUT "***EXIT ?(Y/N)***";S$
220 IF S$="N" THEN CLS : GOTO 10
230 IF S$="Y" THEN GOTO 240 ELSE GOTO 210
240 CLS : STOP

```

Frans HENNAU
 Membership 000692
 Sparrenstraat 5
 B-3110 ROTSELAAR
 BELGIUM

PROGRAM LISTING



MOUSE

This is a 'mouse' menu routine that may be used several times in a program, with a different number of choices on each occasion if necessary. Lines 100 to 190 are the actual routine. The value assigned to OPT in line 1000 is the number of options to be included in the menu, and must not exceed 20. If there are 10 or fewer options they will be displayed double spaced on the screen. If between 10 and 21 they will be single spaced.

The array for the menu options must be set up early in the program (line 10); it is emptied by lines 1005 to 1020, and re-assigned in lines 1030 to 1220. Lines 160 to 190 set up and move the pointer, and the HOME key is used to select the option. The output of the routine is the number of the chosen option, as the value of CH.

Lines 200 to 240 are a general purpose confirmation subroutine that may be used in three different ways:

1. If entered at line 200 the prompt will be displayed at the foot of virtual screen 5.
2. If entered at line 210 the prompt must be positioned by a CSR command included in the main program or follow a ';'.
3. If entered at line 220 it will elicit a yes/no response to some other prompt if included in the main program.

The output from this subroutine is a value for YN, '1' for yes, '0' for no. ★



THE PERSON WHO WROTE THIS NIFTY LITTLE PROGRAM FORGOT TO MENTION HIS NAME.



```
10 DIM MENU$(20,35)
20 GOTO 1000
100 VS 5: CLS : PRINT "Select:"
110 FOR X=1 TO OPT
120 CSR 2,2+X: PRINT X;" ";MENU$(X)
130 NEXT X
140 IF OPT<11 THEN LET A=2
150 IF OPT>10 THEN LET A=1
155 CSR 0,2+A*OPT: PRINT "Use cursor control keys, then HOME"
160 LET CY=2: LET TY=2: CSR 1,CY: PRINT ":"
165 IF ASC(INKEY$)=10 THEN LET TY=TY+A: IF TY>A*OPT THEN LET TY=A
170 IF ASC(INKEY$)=11 THEN LET TY=TY-A: IF TY<A THEN LET TY=A*OPT
175 CSR 1,CY: PRINT ":" : CSR 1,TY: PRINT ":"
180 LET CY=TY
185 IF ASC(INKEY$)=26 THEN LET CH=TY/A: RETURN
190 PAUSE 100: GOTO 165
```

```
200 CSR 2,23
210 PRINT "OK? y/n ";
220 IF INKEY$="Y" OR INKEY$="y" THEN LET YN=1: PRINT "y": RETURN
230 IF INKEY$="N" OR INKEY$="n" THEN LET YN=0: PRINT "n": RETURN
240 GOTO 220
1000 LET OPT=5
1005 FOR X=1 TO OPT
1010 LET MENU$(X)=""
1020 NEXT X
1030 LET MENU$(1)="Name of option 1"
1040 LET MENU$(2)="Name of option 2"
1050 LET MENU$(3)="Name of option 3"
1060 LET MENU$(4)="Name of option 4"
1070 LET MENU$(5)="Name of option 5"
1230 GOSUB 100
1240 CSR 0,2+A*OPT: PRINT "
1250 CSR 0,2+A*OPT: PRINT "Option";CH;" selected. ";: GOSUB 210
1260 IF YN=0 THEN GOTO 1230
```



UTILITY **HEX DUMP**

```

10 REM*****
20 REM*      Hexadecimal memory dump      *
30 REM* (C) 4.11.84 MEMBRAIN Software *
40 REM*      S.Varley                      *
50 REM* Written for MEMOPAD                *
60 REM* the official magazine of          *
70 REM*      GENPAT                      *
80 REM*****
90 REM
100 LPRINT CHR$(27);"E";: REM EMPHASIS ON
110 LET START=0: REM START FROM
120 LET END=60: REM FINISH AT
130 LET COLUMNS=24: REM NUMBER OF COLUMNS ACROSS PRINTER
140 LET GAP$=" "
150 LPRINT CHR$(27);"-";CHR$(1);: REM UNDERLINE
160 LPRINT "HEX dump from ";
170 LET H=INT(START/256): LET L=MOD(START,256)
180 GOSUB 360
190 LPRINT "H to ";
200 LET H=INT(END/256): LET L=MOD(END,256)
210 GOSUB 360
220 LPRINT "H"
230 LPRINT CHR$(27);"-";CHR$(0);: REM UNDERLINE OFF
240 FOR A=START TO END
250 IF MOD(A-START,COLUMNS)=0 THEN LPRINT : LET H=INT(A/256):
LET L=MOD(A,256): GOSUB 360: LPRINT " ";
260 LET N=PEEK(A)
270 LET Z=INT(N/16): GOSUB 330: LET Z=MOD(N,16): GOSUB 330
280 LPRINT GAP$;
290 NEXT A
300 LPRINT CHR$(27);"F": REM EMPHASIS OFF
310 STOP
320 REM CONVERT DECIMAL DIGIT TO HEX
330 IF Z>9 THEN LPRINT CHR$(Z+55);
340 IF Z<=9 THEN LPRINT CHR$(Z+48);
350 RETURN
360 LET Z=INT(H/16): GOSUB 330
370 LET Z=MOD(H,16): GOSUB 330
380 LET Z=INT(L/16): GOSUB 330
390 LET Z=MOD(L,16): GOSUB 330
400 RETURN

```

Example run...

HEX dump from 0000H to 003CH

```

0000 F3 AF 21 00 40 C3 94 01 5E 23 56 23 C9 FF FF FF E3 F5 7E FE
0018 C3 74 3B D7 2D 0A C9 00 33 33 C2 40 3C C3 38 3C 22 6F FD E1
0030 E5 DD E5 FD E5 C3 CC 24 C3 4E FD F5 E3

```



Competition

ONE LINER

This one-liner from D Bowles should help members when setting up the colour on a TV or monitor.

```
10 VS 4: COLOUR 4,1: PAPER 1: CLS: LET I=1: FOR X=16 TO 239 STEP 16: LET
I=I+1: INK I: FOR Y=0 TO 191: LINE X,Y,X+15,Y: NEXT: NEXT
20 GOTO 20
```

This is a one-liner from Gary Wilkinson

```
10 LET G$="": CLS: INPUT "ENTER A WORD ";A$: LET X=LEN (A$): FOR T=X TO 1
STEP -1: LET B$=MID$(A$,T,1): LET G$=G$+B$: NEXT T: PRINT "REVERSED WORD
IS ";G$
```

A couple of one-liners from Eric Roy

[Decimal to Binary - 0 to 255]

```
10 CLS: INPUT "Decimal No ";D: LET B$="": FOR R=9 TO 1 STEP -1: LET B=D-
INT(D/2)*2: LET B$(R)=CHR$(B+48): LET D=INT(D/2): NEXT: PRINT B$(2,9)
```

[Binary to Decimal]

```
10 CLS: INPUT "Binary No ";B$: LET D=0: LET C=-1: LET N=LEN (B$)+1: FOR
R=1 TO 2 STEP 0: LET C=C+1: LET N=N-1: IF N<1 THEN PRINT D ELSE IF
B$(N)="0" THEN NEXT ELSE LET D=D+2^C: NEXT
```

A one liner from T Munson which lists the function key keywords

```
1 LET K$="REMCLSASSAUTBAUVS CONUSECVSCLRCLKATTCOLINKCSRDAT":FOR I=1 TO
8:CSR 16,18: PRINT K$(3*I-2,3),"F";I,K$(3*I+22,3): PAUSE 2000: NEXT I:
GOTO 1
```

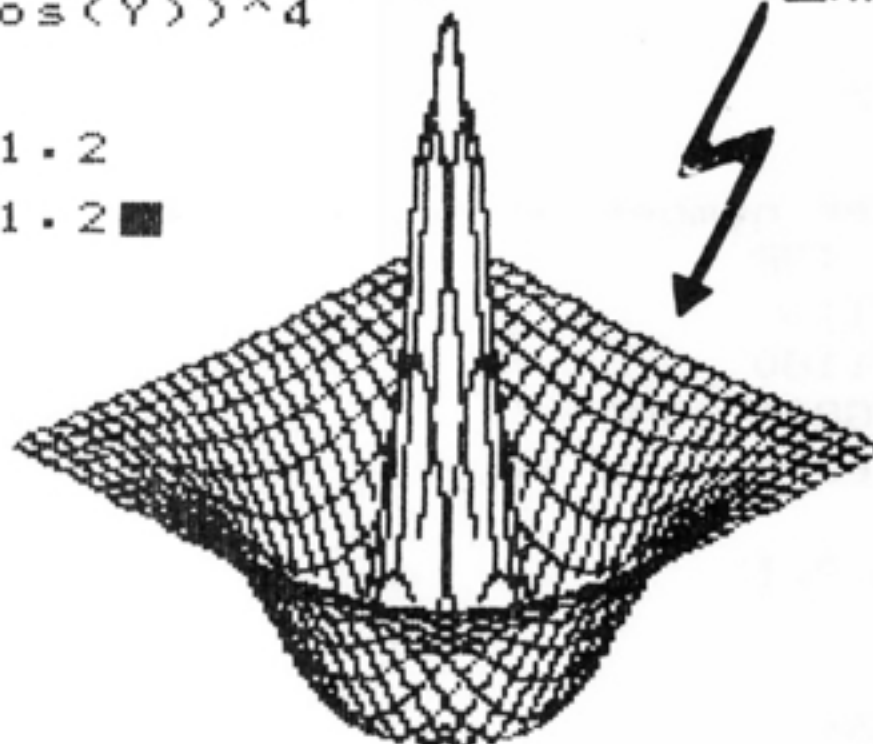
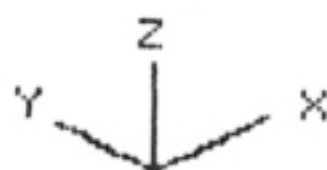
The one liner: Mr I.A. Nichols.

```
1 LPRINT CHR$(27);"A";CHR$(8): VS 4: FOR Y=191 TO 6 STEP -8:
LPRINT CHR$(27);"K" ;CHR$(0);CHR$(1);: FOR X=0 TO 255:
LET A$=GR$(X,Y,8): LPRINT A$;: NEXT : LPRINT : NEXT
```

$$Z = (\cos(X) * \cos(Y))^{30} - (\cos(X) * \cos(Y))^4$$

$$-1.2 \leq X \leq 1.2$$

$$-1.2 \leq Y \leq 1.2$$



PROGRAM LISTING



I am a Student at Trent Polytechnic and am on a BSc Mechanical Engineering degree course. As part of this course we do quite a lot of practical laboratory work (stress analysis, thermodynamics etc.). The results obtained often have to be expressed graphically, and these often turn out to be straight line graphs. I thought, therefore, that it would be useful if the 'best' straight line could be found mathematically, resulting in minimum errors. This is not difficult to do, using a method of linear regression, the principle of 'least squares straight line fit'. This then is the basis of the enclosed program.

The program title is "GRAPH" and is, I hope self explanatory. There are five input prompts and are, in order;

The number of points to be plotted (a single number between 2 and 10, 0 ends)

The x,y coordinates (2 numbers on one line separated by a comma)

A prompt for a graphical display (enter Y or y)

The first and last x coordinates for the best straight line (2 numbers separated by commas)

A prompt to return to the start

The maximum number of points is set at 10 but can be adjusted as required. The graph can also be extrapolated by entering values for the straight line which are outside the range of x values previously entered.

The way that the data is displayed is such that for totally positive (or totally negative) graphs the origins have been suppressed. That is the graph will fill the display area. This could be altered if the relationship with the true origin was required. Graphs which span the origin have the true origin shown (obviously!). A printed copy of the graphs can be produced using a screen dump, such as the one available from Membrain Software, which was used to produce the accompanying examples. To dump the graphics screen to the printer simply exit from the program after displaying the graph (enter 0 to end), then use the screen dump as instructed.

```

10 REM*****
20 REM*          DATA ANALYSIS          *
30 REM*    STRAIGHT LINE GRAPH PLOTS    *
40 REM*LEAST SQUARES STRAIGHT LINE FIT*
50 REM*          *
60 REM*    D. ANDERSON    DEC. 1984      *
70 REM*          *
80 REM*    WRITTEN ON MTX 500            *
90 REM*****
100 DIM A(2,10),SL(10)
110 LET SX=0: LET SY=0
120 LET SXX=0: LET SXY=0
125 LET CFX=0: LET CFY=0
130 CLS
140 CSR 2,1: INPUT "Enter number of points ";N
145 IF N=0 THEN CLS : STOP
150 IF N<=1 THEN GOTO 1150
155 IF N>10 THEN GOTO 1180
160 IF N<>INT(N) THEN GOTO 1090
170 CSR 2,3: PRINT "Enter values (x,y)"
180 FOR I=1 TO N
190 INPUT " ";A(1,I),A(2,I)
200 LET SX=SX+A(1,I)
210 LET SY=SY+A(2,I)
220 LET SXX=SXX+A(1,I)*A(1,I)
230 LET SXY=SXY+A(1,I)*A(2,I)
240 NEXT I

```



```
250 LET AN=N
260 LET B=(SX*SY-AN*SXY)/(SX*SX-SXX*AN)
270 LET A=(SY-B*SX)/AN
280 LET A=(INT(A*10000))/10000
290 LET B=(INT(B*10000))/10000
300 CLS
310 IF B>0 THEN LET C$="+" ELSE LET C$=""
320 CSR 4,6: PRINT "The equation is : "
330 CSR 4,8: PRINT "Y=";A;C$;B;"X"
340 CSR 2,23: INPUT "Do you want the graph displayed ? ";G$
350 IF G$="Y" OR G$="y" THEN GOTO 450 ELSE GOTO 110
449 REM
450 REM*****graph display*****
451 REM
460 CLS
470 CSR 1,2: INPUT "Enter first and last x values ";X1,X2
480 CRVS 2,1,0,0,32,24,32
490 VS 2
500 PAPER 4: COLOUR 4,4
510 CLS
514 REM
515 REM*****sort array*****
516 REM
520 FOR I=N-1 TO 1 STEP -1
530 LET T=A(1,I)
540 LET T2=A(2,I)
550 FOR J=I+1 TO N
560 IF T<A(1,J) THEN GOTO 600
570 LET A(1,J-1)=A(1,J)
580 LET A(2,J-1)=A(2,J)
590 NEXT J
600 LET A(1,J-1)=T
610 LET A(2,J-1)=T2
620 NEXT I
624 REM
625 REM*****find xscale*****
626 REM
630 IF X1<=A(1,1) THEN LET XK1=X1 ELSE LET XK1=A(1,1)
640 IF X2>=A(1,N) THEN LET XK2=X2 ELSE LET XK2=A(1,N)
641 IF XK1>0 AND XK2>0 THEN LET XSCALE=200/(XK2-XK1): GOTO 645
643 IF XK1<0 AND XK2<0 THEN LET XSCALE=200/(ABS(XK1)-ABS(XK2)): GOTO 645
644 LET XSCALE=200/(ABS(XK1)+ABS(XK2))
645 IF XK1<0 THEN LET CFX=(ABS(XK1))*XSCALE
646 IF XK1>0 THEN LET CFX=-XK1*XSCALE
649 REM
650 REM*****find yscale*****
651 REM
660 FOR I=1 TO N
670 LET SL(I)=A(2,I)
680 NEXT I
690 FOR I=N-1 TO 1 STEP -1
700 LET T=SL(I)
710 FOR J=I+1 TO N
720 IF T<SL(J) THEN GOTO 750
730 LET SL(J-1)=SL(J)
740 NEXT J
750 LET SL(J-1)=T
760 NEXT I
770 LET Y1=X1*B+A: LET Y2=X2*B+A
775 IF B<0 THEN GOTO 795
```

```

776 REM
777 REM*****+ve grad*****
778 REM
780 IF Y1>=A(2,1) THEN LET YK1=A(2,1) ELSE LET YK1=Y1
785 IF A(2,N)<=Y2 THEN LET YK2=A(2,N) ELSE LET YK2=Y2
786 IF YK1>0 AND YK2>0 THEN LET YSCALE=150/(YK2-YK1): LET YKC=YK1: GOTO 815
787 IF YK1<0 AND YK2<0 THEN LET YSCALE=150/(ABS(YK1)-ABS(YK2)):
LET YKC=YK1: GOTO 815
790 LET YKC=YK1
791 GOTO 810
794 REM
795 REM*****-ve grad*****
796 REM
800 IF Y1<=A(2,1) THEN LET YK1=A(2,1) ELSE LET YK1=Y1
802 IF A(2,N)>=Y2 THEN LET YK2=Y2 ELSE LET YK2=A(2,N) GOTO 816
803 IF YK1>0 AND YK2>0 THEN LET YSCALE=150/(YK1-YK2): LET YKC=YK2:
804 IF YK1<0 AND YK2<0 THEN LET YSCALE=150/(ABS(YK1)-ABS(YK2)):
LET YKC=YK2: GOTO 815
805 LET YKC=YK2
810 LET YSCALE=150/(ABS(YK1)+ABS(YK2))
815 IF YKC<0 THEN LET CFY=(ABS(YKC))*YSCALE
816 IF YKC>0 THEN LET CFY=-YKC*YSCALE
819 REM
820 REM*****scale values*****
821 REM
830 FOR I=1 TO N
840 LET A(1,I)=A(1,I)*XSCALE+CFX
850 LET A(2,I)=A(2,I)*YSCALE+CFY
860 NEXT I
865 LET X1=X1*XSCALE+CFX
866 LET X2=X2*XSCALE+CFX
867 LET Y1=Y1*YSCALE+CFY
868 LET Y2=Y2*YSCALE+CFY
869 REM
870 REM*****plot st. line*****
871 REM
880 LINE X1+30,Y1+10,X2+30,Y2+10
889 REM
890 REM*****plot points*****
891 REM
900 FOR I=2 TO N
910 LINE A(1,I-1)+30,A(2,I-1)+10,A(1,I)+30,A(2,I)+10
920 NEXT I
929 REM
930 REM*****print equation*****
931 REM
940 CSR 6,2: PRINT "Y=";A;C$;B;"X"
949 REM
950 REM*****draw axes*****
951 REM
955 IF CFX<0 THEN LET CFX=0
956 IF CFY<0 THEN LET CFY=0
957 IF CFX>200 THEN LET CFX=200
958 IF CFY>150 THEN LET CFY=150
960 PHI 0
970 PLOT 30+CFX,160
980 ANGLE 3*PI/2
981 DRAW 150
982 FOR I=30+CFX TO 230 STEP 5

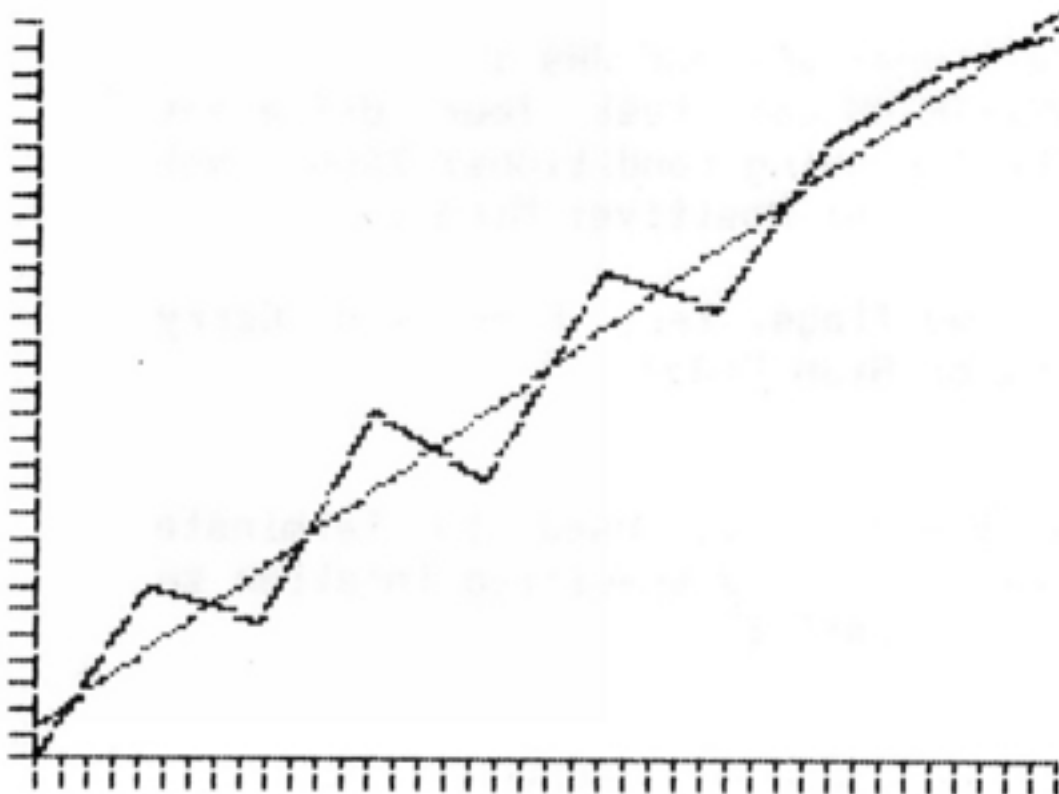
```

```

983 PLOT I,9+CFY
984 DRAW 5
985 NEXT I
986 FOR I=30+CFX TO 30 STEP -5
987 PLOT I,9+CFY
988 DRAW 5
989 NEXT I
990 PLOT 30,10+CFY
1000 PHI PI/2
1010 DRAW 200
1020 FOR I=20+CFY TO 160 STEP 5
1030 PLOT 25+CFX,I
1040 DRAW 5
1050 NEXT I
1051 FOR I=20+CFY TO 10 STEP -5
1052 PLOT 25+CFX,I
1053 DRAW 5
1054 NEXT I
1060 INPUT " ";ST$
1075 VS 5
1080 GOTO 110
1089 REM
1090 REM*****error messages*****
1091 REM
1100 CLS
1110 CSR 2,4: PRINT "N MUST BE AN INTEGER!"
1120 CSR 13,12: PRINT "W A I T !"
1130 PAUSE 5000
1140 GOTO 110
1150 CLS
1160 CSR 2,4: PRINT N;" IS TOO FEW ARGUMENTS!"
1170 GOTO 1120
1180 CLS : CSR 2,4: PRINT "N MUST BE <=10!"
1190 GOTO 1120

```

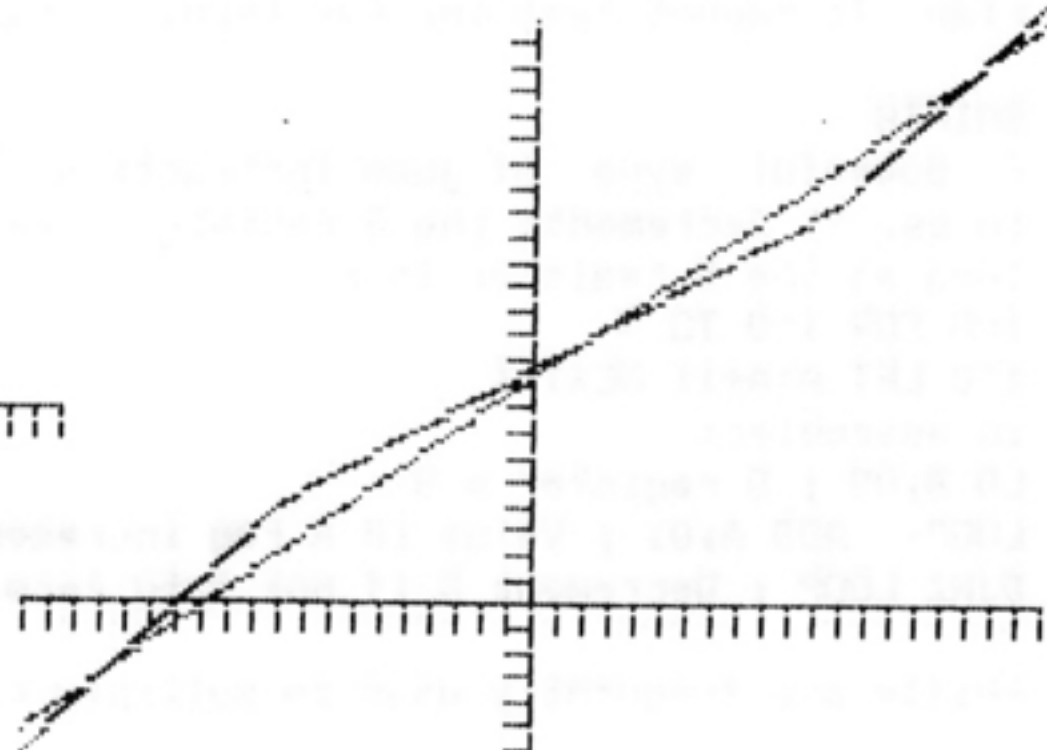
$$Y = 4.8545 + 1.1454X$$



David Anderson

David Anderson

$$Y = 4.5743 + 0.7027X$$



ASSEMBLY LANGUAGE

Jump instructions cause normal, sequential program execution to branch to a different part of the program. There are basically two types: **JP, 3C00 Hex** = absolute jump, and **JR, START** = relative jump. These instructions can be split into two further groups:

- * The unconditional jump, which causes a branch to another part of the program regardless of any conditions; and

- * The conditional jump, which forces a branch to another segment of the program if one or more stipulated conditions are met.

In the example in figure 1, **NEXT** can be located at any address within the range of -128 to +127 bytes from the current instruction **LD A, (BUFFER)**. The relative jump saves one byte over the **JP** instruction, and code constructed using **JR** type jumps is relocatable. The assembler will calculate the correct displacement for you and insert it in the object on assembly.

Conditional Jump

BASIC : LET A=X: IF X=3 THEN GOTO 1500

ASSEMBLER : LD A, (BUFFER) ;get value from buffer
 CP 3 ;is A=3
 JP Z, FINISH ;yes then goto FINISH

Relative Unconditional Jump

BASIC : LET A=X: GOTO 1500

ASSEMBLER : LD A, (BUFFER)
 JR NEXT ;jump to segment labelled NEXT

If you use a **JR** instruction illegally the assembler will point out the error.

There are some differences between conditional **JPs** and **JRs**:

- * The conditional absolute **JP**: **JP NC, FINISH** can test four different flags, and a branch can be forced of the following conditions: Zero: Not Zero: Carry: No Carry: Parity Odd: Parity Even: Positive: Minus.

- * The relative jump can test for only two flags, Zero Flag and Carry Flag. It cannot test the Parity/overflow or Sign Flags.

SHIFTS

A powerful type of jump instruction, the **DJNZ**, is used to terminate loops. It decrements the B register then jumps to a specified location so long as the B register is not zero (NZ). In Basic:

100 FOR I=0 TO 9

110 LET A=A+1: NEXT I

In assembler:

LD B, 09 ; B register = 9

LOOP: ADD A, 01 ; Value in A reg incremented by 1

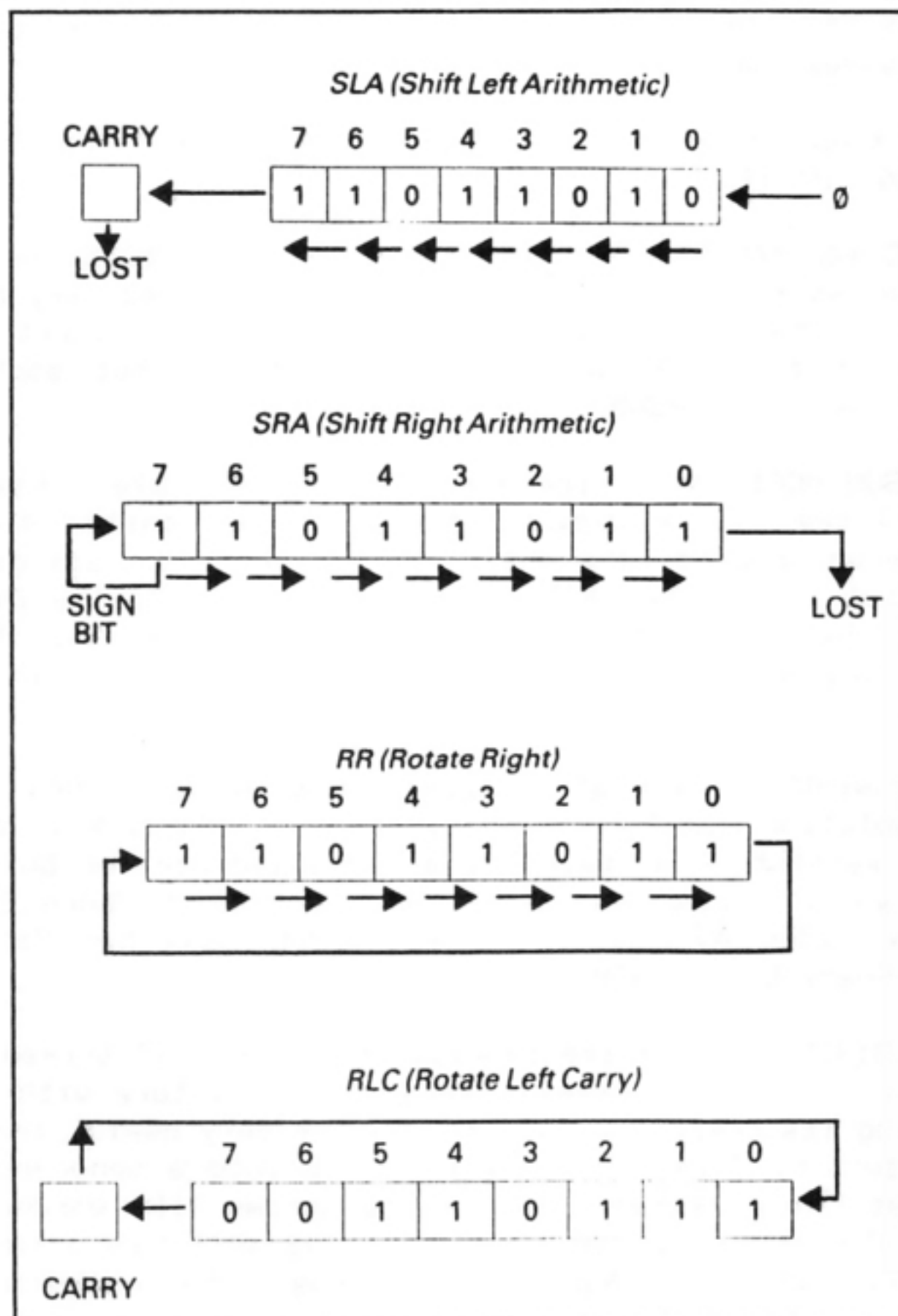
DJNZ LOOP ; Decrement B if not = to zero, go do it all again.

Shifts are frequently used to multiply or divide by the power of two. For

instance, suppose an eight-bit number is shifted right by one bit position. The result is the same as dividing by two, with the remainder being discarded. On the other hand, shifting an eight-bit number left by one bit position is the same as multiplying by two.

There are two shift instructions to move bits to the right - one arithmetic shift and one logical shift. The **SRL** (Shift Right Logical) operates in exactly the same way as the **SLA** instruction with the carry bit being lost and a zero being moved into the seventh bit. However, the **SRA** (Shift Right Arithmetic) preserves the sign bit 7.

Rotates are identical in operation to shifts except that the seventh bit, which would normally be lost, is rotated and preserved. ★



Competition

A one liner from T Munson which lists the function key keywords
 1 LET K\$="REMCLSASSAUTBAUVS CONUSECVSCLRCLKATTCOLINKCSRDAT":FOR I=1 TO
 8:CSR 16,18: PRINT K\$(3*I-2,3),"F";I,K\$(3*I+22,3): PAUSE 2000: NEXT I: PAGE 17 .
 GOTO 1

HARDWARE REVIEW MONITOR ROUNDUP



The **MICROVITEC CUB** is one of the best known computer monitors due to its popularity in the educational field. The one we use here at Genpat, on one of our FDX systems, is the 653 model in a black plastic case (it is also available in a metal case) with an RGB only input. A version is available with composite and sound inputs.

The Cub has a video bandwidth of 18MHz, which gives it a medium resolution classification, with a dot pitch of 0.43mm and a horizontal pixel count of 653. This monitor can be recommended for those of you who have an FDX system and are looking for a high quality product. The price is #375.

More details from: Microvitec plc, Futures Way, Bolling Rd, Bradford, West Yorkshire, BD4 7TU

The **MICROVITEC 1431/AP/MS** available from Genpat at #225. We use it with an MTX when reviewing or developing software. It gives very good results and is useful to Memotech owners as it has an audio input socket. The other version of this monitor with just an RGB input socket is also available from Genpat at #205.

The new **FERGUSON MC01** looks like a normal 14" portable colour TV but is in fact the latest in a growing number of TV/monitor to have been launched in recent months. The set has eight buttons - six giving normal TV reception but one marked RGB and the other VIDEO. The input sockets provided are the standard aerial, RGB and composite. The monitor automatically switches between the types of input when you select the channels.

We have no bandwidth or resolution figures but the MC01 uses the standard TV tube so resolution could also be described as standard. With this set you don't get the resolution achieved by a dedicated monitor but at #229 it is excellent value for money. More details from : Thorn EMI Ferguson (Service Div), Video Advisory Service, PO box 121, Lea Valley Trading Estate, Angel Road, London N18.

The **SANYO CD 3195B** has TV-like resolution on its 14" screen with a dot pitch of 0.5mm. The screen gives a very stable picture with good colour saturation using its composite only input. One very useful feature is the green screen switch. This turns the CD 3195B into a monochrome monitor, providing greater eye relief when using programs like wordprocessors or spreadsheets. Overall it is an attractive and well-made monitor, but is rather expensive at #360. More details from : Sanyo Marubeni UK Ltd, Otters Pool Way, Watford, Herts.

We would like to hear any comments from members on their experiences with both colour and monochrome monitors so that we can give other members advice on which monitors are best for their MTX or RS.





SYNTAXSOFT Presents

HUNCHY

The Challenge !!

4.95

MEMOTEXT

THE ULTIMATE TAPE BASED
WORD PROCESSOR

9.95

INCLUDING A 32 PAGE
INSTRUCTION MANUAL.

TAPE-TO-DISC

TRANSFER

THIS IS THE ONE ALL YOU
DISC OWNERS HAVE BEEN
WAITING FOR.....

TRANSFER YOUR TAPES TO
DISC.... WILL ALSO
WORK ON NEWWORD FILES.

6.95

SUPPLIED ON DISC
INCLUDING P & P

PROGRAM LISTING

joystick

We have lost the name of the person who sent the following program, so could they please tell us who they are.

This program scans the joystick ports 1 and 2 and can be called by the USR function.

```
LET A=USR(16397)      Joystick 1
LET B=USR(16415)      Joystick 2
```

The addresses will need changing for the MTX500 or if the code is moved.

10 CODE

```
4007          RET
4008 TAB:      DB 191,251,239,247,223
400D          LD HL,TAB;      JOYSTICK 1
4010          LD BC,#05FF
4013 LOOP:     LD A,(HL)
4014          OUT (5),A
4016          IN A,(5)
4018          RLA
4019          RL C
401B          INC HL
401C          DJNZ LOOP
401E          RET
401F          LD A,127;      JOYSTICK 2
4021          OUT (5),A
4023          IN A,(5)
4025          LD C,A
4026          IN A,(6)
4028          RRA
4029          RL C
402B          LD A,C
402C          OR 224
402E          LD C,A
402F          LD B,0
4031          RET
```

Symbols:

TAB4008LOOP4013

Competition

This one-liner from D Bowles should help members when setting up the colour on a TV or monitor.

```
10 VS 4: COLOUR 4,1: PAPER 1: CLS: LET I=1: FOR X=16 TO 239 STEP 16: LET
I=I+1: INK I: FOR Y=0 TO 191: LINE X,Y,X+15,Y: NEXT: NEXT
20 GOTO 20
```

PROGRAM LISTING

DOOG

T J SELDON

FIGHT

Here is a program that should delight all members. T.J. says he is not particularly proud of the coding... maybe so, but the end result justifies the typing.


Dog Fight is a difficult 3D 'shoot'em up' game. You are the only defence between your planet and pillaging aliens. The aliens have no defence against you but are nifty movers. As they fly toward you, you must track them down with your **thermal weapon** - this task is not made any easier by the fact that your craft accelerates in a particular direction which cannot be changed instantly as the aliens can.

Your control panel shows which direction the aliens are located when they are out of view, the power of your thrusters in each direction, and your remaining fuel. You may use joystick A or the keypad.


You get points for good tracking - keeping the alien in view - but you lose 5 points every time you fire and miss.

This game is quite difficult to master but for a game written entirely in Basic it is superb ! One point to remember: you are moving your craft not the alien.

This game provides a really good structure for developing an exciting real-time arcade style program. I would like to see members develop this game further by adding features that allow the aliens to blast you etc. Two pieces of software will be awarded to the best improvement on this program. ★



```
1 GOTO 5
4 REM *** GOTO 5 FOR AUTO LOAD ***
5 GOSUB 5000: REM *** GENPATS ***
6 VS 4: CLS
7 REM *** SOUND OFF , SPRITES ON ***
8 FOR I=1 TO 3: SOUND I,0,0: NEXT I
9 CTLSPR 1,4: CTLSPR 2,12: CTLSPR 6,0: GOSUB 200
10 REM *** TITLE SCREEN ***
11 COLOUR 4,7: PAPER 3: INK 1: CLS : GOSUB 4000
15 PAPER 7: CLS
20 REM *** DEFINE SCREEN ***
21 GOSUB 900: REM *** SKY ***
22 GOSUB 1000: REM *** REST ***
23 LET T=250: REM *** FUEL LIMIT ***
24 GOSUB 100: REM *** GET KEY, SHOOT ***
25 GOSUB 300: REM *** ADJUST SAUCER ***
26 REM *** FUEL UPDATE, MOVE SHIP ***
27 LET T=T-1: PAPER 0: INK 1: CSR 17,16: PRINT "    ": CSR 17,16: PRINT T
28 LET YS=YS+(KEY=11 AND YS)-4)-(KEY=10 AND YS<4): LET XS=XS+(KEY=25 AND XS)-4)-(KEY=8 AND XS<4)
29 LET Y=Y+YS*2+YA*1.5: LET X=X+XS*2+XA*1.5: IF SZ=0 THEN LET SZ=1
30 LET A=10: LET B=19: PAPER 3: CSR A+2,B: IF Y<105 THEN LET SZ=0: LET Y=105: PRINT "v" ELSE PRINT " "
```




```

31 REM *** DIRECTION FINDER ***
32 CSR A,B-2: IF X<10 THEN LET SIZ=0: LET X=10: PRINT "<" ELSE PRINT " "
33 CSR A+4,B-2: IF X>215 THEN LET SIZ=0: LET X=215: PRINT ">" ELSE PRINT " "
34 CSR A+2,B-4: IF Y>160 THEN LET SIZ=0: LET Y=160: PRINT "^" ELSE PRINT " "
35 FOR SP=2 TO 4: ADJSR 3,SP,Y: NEXT SP
36 CSR 11,16: PRINT YS*(YS<0): CSR 11,18: PRINT -YS*(YS>0): CSR 12,17: PRINT XS*(XS<0): CSR 10,17: PRINT -XS*(XS>0)
37 IF MOD(T,5)=0 THEN LET Z=SIZ/2: LET XA=Z*(RND<0.5)-Z*(RND<0.5): LET YA=Z*(RND<0.5)-Z*(RND<0.5): LET SIZ=SIZ-1*(SIZ<8): LET SC=SC+INT(SIZ/3)
39 REM *** SCORE, CHECK END ***
40 CSR 17,19: PAPER 0: INK 1: PRINT "   ": CSR 17,19: PRINT SC
50 IF T<1 THEN FOR I=900 TO 1000: SOUND 1,I,7: NEXT I: CSR 5,13: PAPER 1: INK 11: PRINT "GAME OVER-PRESS SPACE BAR": IF INKEY$("<") THEN GOTO 50
60 IF T<1 THEN GOTO 6
90 SOUND 3,7,8: SOUND 2,T,0
99 GOTO 24
100 LET KEY=ASC(INKEY$)
105 IF KEY=26 THEN SOUND 1,1200,240,40,30,10,1 ELSE RETURN
106 LET SC=SC-5
110 LET F=1: PAPER 1: FOR I=4 TO 7: INK 11-4*(I=4)+(I=7): CSR F+9,I: PRINT "k": CSR 22-F,I: PRINT "l"
117 CSR F+9,I: PRINT " ": CSR 22-F,I: PRINT " ": LET F=F+2: NEXT I
118 IF X<112 OR X>126 OR Y<124 OR Y>136 THEN GOTO 140
119 FOR I=32 TO 37: SOUND 3,7,15: SOUND 2,I+RND*10,-1*(MOD(I,2)=0)
120 FOR SP=2 TO 4: ADJSR 1,SP,MOD(I,12)-4*(I<37)+1: NEXT SP
130 NEXT I
133 LET X=RND*205+10: LET SIZ=1
135 LET SC=SC+25*SIZ
140 SOUND 1,0,0: RETURN
200 LET SIZ=1: LET X=116: LET Y=146: LET XS=0: LET YS=0: SPRITE 2,2,X,Y,0,0,7
201 SPRITE 3,3,X+8,Y,0,0,7
202 SPRITE 4,4,X+16,Y,0,0,7
204 SPRITE 1,1,126,131,0,0,9
210 LET H=0: LET SC=0
220 LET XA=SIZ*(RND<0.5)-SIZ*(RND<0.5): LET YA=SIZ*(RND<0.5)-SIZ*(RND<0.5)
250 RETURN
300 LET SIZ=INT(SIZ): IF SIZ=1 THEN ADJSR 0,2,0: ADJSR 0,3,15: ADJSR 0,4,0: LET C=12
301 IF SIZ=2 THEN ADJSR 0,2,0: ADJSR 0,3,14: ADJSR 0,4,0: LET C=12
302 IF SIZ=3 THEN ADJSR 0,2,0: ADJSR 0,3,13: ADJSR 0,4,0: LET C=4
303 IF SIZ=4 THEN ADJSR 0,2,0: ADJSR 0,3,12: ADJSR 0,4,0: LET C=4
304 IF SIZ=5 THEN ADJSR 0,2,9: ADJSR 0,3,10: ADJSR 0,4,11: LET C=4
305 IF SIZ=6 THEN ADJSR 0,2,7: ADJSR 0,3,3: ADJSR 0,4,8: LET C=5
306 IF SIZ=7 THEN ADJSR 0,2,5: ADJSR 0,3,3: ADJSR 0,4,6: LET C=7
307 IF SIZ=8 THEN ADJSR 0,2,2: ADJSR 0,3,3: ADJSR 0,4,4: LET C=7
308 IF SIZ=0 THEN ADJSR 0,2,0: ADJSR 0,3,0: ADJSR 0,4,0: LET C=7
310 FOR SP=2 TO 4: ADJSR 1,SP,C: ADJSR 2,SP,X+(SP-2)*8: NEXT SP
320 RETURN
900 CLS : CSR 10,3: PAPER 1: PRINT "   "
902 CSR 10,2: PRINT "   "
903 INK 4: PAPER 7: CSR 9,3: PRINT "i": CSR 22,3: PRINT "j": CSR 7,2: PRINT "inn": CSR 22,2: PRINT "nnj"
904 CSR 7,3: PAPER 1: INK 7: PRINT "i": CSR 24,3: PRINT "j"
906 CSR 3,3: PRINT "   ": CSR 25,3: PRINT "   "
909 FOR I=4 TO 11
910 CSR 1,I: PRINT "   "
920 NEXT I
930 RETURN
1000 INK 5
1002 CSR 1,12: PAPER 4: PRINT "eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee"
1003 CSR 1,13: PRINT "ijijijijijijijijijijijijijijijij"
1004 FOR Q=14 TO 20
1005 CSR 1,Q: PRINT "   "
1006 CSR 23,Q: PRINT "   "
1007 NEXT

```

```

1012 CSR 4,21: PAPER 7: INK 4: PRINT "hhhhhhhhhhhhhhhhhhhh"
1020 LET T=3: FOR I=16 TO 18
1030 CSR T,I: INK 2: PAPER 6: PRINT "j": CSR 31-T,I: PRINT "i": LET T=T+1: NEXT I
1040 CSR 1,16: INK 1: PRINT " ": CSR 29,16: PRINT " ": CSR 1,17: PRINT "j": CSR 30,17: PRINT "i"
1050 CSR 1,3: INK 7: PAPER 4: PRINT "i": CSR 29,3: PRINT "j ": CSR 1,2: PRINT "i": CSR 30,2: PRINT "j"
1060 INK 15: PAPER 12: CSR 17,18: PRINT "SCORE": CSR 17,15: PRINT " FUEL"
1100 INK 1: PAPER 7
1102 REM *** PANEL GRAPHICS ***
1106 LINE 134,50,178,50: LINE 134,51,178,51
1109 LINE 76,26,76,75: LINE 76,75,132,75: LINE 132,75,132,26: LINE 132,26,76,26
1110 LINE 73,24,73,78: LINE 73,78,182,78: LINE 182,78,182,24: LINE 182,24,73,24
1113 LINE 0,191,24,167: LINE 255,191,231,167
1114 LINE 40,184,56,168: LINE 215,184,199,168
1115 LINE 40,184,40,191: LINE 215,184,215,191
1120 CSR 2,18: PAPER 1: INK 5: PRINT "j": CSR 29,18: PRINT "i": CSR 2,19: PRINT "j": CSR 29,19: PRINT "i": CSR 2,20: PRINT "j": CSR 29,20: PRINT "i"
1130 CSR 5,0: PAPER 7: PRINT "hhhhhhhhhhhhhhhhhhhh"
1140 FOR J=1 TO 21 STEP 2
1145 INK 6
1160 LET DE=0: FOR J=167 TO 153 STEP -2
1170 LINE 8+DE,J+16,40+DE,J+16: LINE 247-DE,J+16,215-DE,J+16: LET DE=DE+2: NEXT J
1200 FOR I=15 TO 19
1210 CSR 10,I: PAPER 3: PRINT " ": NEXT I
1400 RETURN
4000 SOUND 1,PEEK(I)+100,10: SOUND 2,PEEK(I-MOD(I,3))+250,9
4001 PAPER (MOD(I,16)+1)
4002 CSR 10,9: PRINT " DOG FIGHT "
4003 CSR 3,11: PRINT "BY T.J.S. GRAPHICS BY A.G.M"
4005 INK 1: IF INKEY$="" THEN LET I=I+1: GOTO 4000
4006 PRINT **: SOUND 1,0,0: SOUND 2,0,0
4007 RETURN
5000 GENPAT 0,101,0,255,0,0,255,255,255,0
5010 GENPAT 0,104,0,0,90,255,189,231,195,195
5020 GENPAT 0,105,255,127,63,31,15,7,3,1
5030 GENPAT 0,106,255,254,252,248,240,224,192,128
5040 GENPAT 0,107,240,240,240,248,24,4,2,1
5050 GENPAT 0,108,15,15,15,31,24,32,64,128
5060 GENPAT 0,109,255,255,255,255,255,255,255,255
5070 GENPAT 3,0,0,0,0,0,0,0,0,0
5080 GENPAT 3,1,255,129,0,0,0,0,129,255
5090 GENPAT 3,2,3,15,63,255,255,63,15,3
5100 GENPAT 3,3,255,255,255,255,255,255,255,255
5110 GENPAT 3,4,224,240,252,255,255,252,240,224
5120 GENPAT 3,5,1,7,15,63,63,15,7,1
5130 GENPAT 3,6,128,224,240,252,252,240,224,128
5140 GENPAT 3,7,0,3,7,31,31,7,3,0
5150 GENPAT 3,8,0,192,224,248,248,224,192,0
5160 GENPAT 3,9,0,0,1,3,3,1,0,0
5170 GENPAT 3,10,0,126,255,255,255,255,126,0
5180 GENPAT 3,11,0,0,128,224,224,128,0,0
5190 GENPAT 3,12,0,0,60,255,255,60,0,0
5200 GENPAT 3,13,0,0,24,126,126,24,0,0
5210 GENPAT 3,14,0,0,0,60,60,0,0,0
5220 GENPAT 3,15,0,0,0,24,0,0,0
5230 GENPAT 3,16,0,0,0,24,0,0,0
5240 RETURN
60000 PRINT ASC(INKEY$): GOTO 60000

```



ONE LINK

```

10 REM I. MIDWINTER
20 REM SPRITE DEMO
30 REM USE ABBREVIATED KEYWORDS
40 VS 4: COLOUR 2,1: CLS : CTLSPR 2,32: CTLSPR 0,1: CTLSPR 6,1: CTLSPR 5,32: CTLSPR 3,32: GENPAT 3,0,1,6,254,124,60,252,108,36: FOR N=1 TO 32: LET A=RND*99+9:
SPRITE N,0,0,0,A+RND,A+RND,RND*14+2: NEXT : PAUSE 6E4: GOTO 40

```

REFERENCE

THE COMPLETE

DATA SAVE & LOAD

ANSWER ?

We are publishing here a very important document: the instructions to a program, we believe, solves the problem of Data Save & Load. As most members are aware, this has caused great difficulty to people who own MTX 512's because the internal save mechanism would not automatically switch pages. This program is the answer.

The coding is quite long and has been written using Microsoft's M.80, therefore, in order not to cause confusion, for those of you who want to get their 'hands dirty' and use the coding we will supply the full listing for 1.00. Members who would like the tape, which you simply load in, run, and forget, can obtain it for 2.50 inclusive. Send in to the club but please allow 7 days for delivery.

Variable Saving, Loading & Verifying with Tape Cassette

This document describes the cassette program "VTAPE", which contains routines for saving, loading and verifying MTX BASIC string, character array, simple numeric and numeric array variables on cassette.

1. Description

The program "VTAPE" consists of one short machine-code routine which, when run, copies the variable tape routines into high memory. It then resets the BASIC program pointers and returns to BASIC - in other words the program deletes itself. The starting address for the tape routines in high memory is #F000, and the routines end at #F341. This finishing address is well below where the sound buffers are created, so SBUF commands will operate normally.

The tape routines are divided into six sections, and are listed below in the order in which they occur in memory:

- (i) Interface section
- (ii) Save routine
- (iii) Load routine
- (iv) Verify routine
- (v) Miscellaneous routines and buffers
- (vi) Low-level cassette routines

The Interface section is the only one of any importance to the user (details are given below). The routines in the other sections can be assumed to work correctly and so how they work or what they do is of no relevance here.

Saving, loading or verifying variables is done in one of two ways: the routines can be called directly in machine-code, or the BASIC command USER may be used.



BRUNNING SOFTWARE

BRUNWORD £16-57 inc. simply the best!

Can we say more than Keith Hook in issue 2 of GENPAT. "I was amazed how much power the programmer has managed to cram into the tape package..." BrunWord loads in 31 seconds (no wonder he was surprised), has full text editing with INSERT and OVERWRITE, has temporary files in memory, cut & paste, search, block delete, format preview, enlarged, condensed, underline, and emphasized printing and uses a compressed format for memory files and cassette files. This programme was specially written for the 40 column Memotech screen. Address on the right, Dear Sir on the left, yours sincerely in the middle. Type in using 40 columns. Print out any width up to 130 columns without re-formatting. The programme works it out.

DATAFILE £12-80 inc. New release.

Powerful card filing system. Very fast alphabetical, numerical or date search. User defined headings, print formats, card colour.

Both programmes are designed to work with any printer that operates from the parallel printer port. Prices are special GENPAT prices. (Overseas add £2-00 per order.)

Send Cheque or P/O to :-



BRUNNING SOFTWARE,
34 Helston Road,
Chelmsford,
Essex. CM1 5JF
Tel 0245 252854.

2. The Interface Section

The Interface section is situated at the start of the tape code at #F000 and consists of three machine-code jump instructions followed by a routine (see overleaf).

```
#F000: JP VSAVE      ;Jump to variable
saving routine
#F003: JP VLOAD      ;Jump to variable
loading routine
#F006: JP VVERIF     ;Jump to variable
verifying routine
#F009: <entry point from USER command>
```

3. Using the USER command

The USER bytes at #FAB9, #FABA and #FABB should be patched to jump to the tape USER entry routine at #F009. (This patching is most easily done in PANEL.) The syntax bytes at #FAB7 and #FAB8 must also be changed to allow USER commands to be entered without giving a syntax error. The values that these five locations should have is as follows:

```
#FAB7 -> #C9
#FAB8 -> #07
#FAB9 -> #C3
#FABA -> #09
#FABB -> #F0
```

The USER command can now be used to save, load or verify variables.

(i) Saving

Syntax: USER SAVE,<list of variable names>.

e.g.

```
10 LET A$="Hallo!"
20 LET X=999
30 USER SAVE,A$,X
```

This program saves the string A\$ on tape giving it the name "A\$", followed by the number X with the name "X". (Note. The two variables are saved as separate blocks of data on tape, each preceded by a header containing the variable name, type and size. The maximum length of a variable name is 12 characters.)

With numbers, five data bytes are saved (floating-point notation). With strings, the number of bytes saved is the current length of the string.

If the length is zero, then no data bytes are written to tape, just the string header

e.g.

```
10 LET A$="Hello!"
20 REM Dimension of A$ = 64 chars, length = 6
30 DIM B$(8)
40 REM Dimension of B$ = 8 chars, length = 0
50 USER SAVE,A$,B$
60 REM Saves 6 bytes for A$, 0 bytes for B$
```

For both numeric and character arrays, the number of bytes saved is equal to the total size of the array. To indicate that a variable to be saved is an array, a '(' character (but nothing else) must follow the variable name in the USER SAVE command.

e.g.

```
10 LET X=999
20 DIM X(10,10)
30 LET X(5,5)=55
40 USER SAVE,X
50 REM Save number X (5 bytes)
60 USER SAVE,X(
70 REM Save numeric array X (5*5*5 bytes)
```

The type of variables to be saved can be mixed at will:

e.g.

```
10 DIM A$(8),B$(2,2),Y(10),Z(3,3,3)
20 LET A$="abcdef"
30 LET B$(1,1)="?"
40 FOR X=1 TO 10:LET Y(X)=X*X:NEXT
50 LET Z(1,3,2)=56.9049
60 USER SAVE,A$,B$(,X,Y,Z(
```

(ii) Loading

Syntax: USER LOAD,<list of variable names>.

The load routine works in a similar fashion to the SAVE routine. One important point to remember is that before a variable can be loaded from tape it must exist, otherwise the BASIC 'Undefined' error occurs (this applies to saving too!)

e.g.

```
10 LET A$=""
20 REM String A$ now exists
30 USER LOAD,A$
40 PRINT A$
```



In the above example, the tape is searched until a string "A\$" is found. Thus, in order to load data into variable A\$ there must be a variable A\$ saved on tape. However, this is not the only necessary condition for loading a variable. A check is made on the size of the variable on tape and if this size is larger than the maximum size of the variable in memory the variable is NOT loaded. (Instead of loading the variable, this 'variable too large' condition is flagged by storing the value 2 in the BASIC system variable VERIF at \$FD67 or 64871 decimal.)

e.g.

```
10 LET A$="Blah blah blah"
20 USER SAVE,A$
30 REM 14 bytes saved
40 CLEAR
50 DIM A$(10)
60 USER LOAD,A$
70 IF PEEK(64871)=2 THEN PRINT "Variable
```

too big"

In the above program, the string A\$ saved in line 20 is too large to load back in line 60. This size checking takes place for numeric and character arrays as well as for strings, but is not needed for simple numbers.

When a string is loaded, the length of the string is equal to the number of data bytes read from tape, which in turn is simply the length of the string when it was saved.

e.g.

```
10 LET A$=""
20 REM Length of A$ = 0, dimension of A$ = 64
40 USER SAVE,A$
50 REM 0 data bytes saved
60 LET A$="XYZ"
70 PRINT A$,LEN(A$)
80 REM Length of A$ = 3
90 PRINT "Stop tape and rewind"
100 USER LOAD,A$
110 PRINT A$,LEN(A$)
120 REM Length of A$ = 0
```

(iii) Verifying

Syntax: USER VERIFY,<list of variable names>.

The verify routine works in a similar way to the load routine except, of course, that no loading

or changing of existing variables in any manner is done. If a variable has verified correctly the system variable VERIF (see above) contains the value 0, otherwise it contains the value 1. (Note. The 'variable too large' condition is tested in the verify routine, so VERIF may contain the value 2, in which case no verifying is attempted.)

e.g.

```
10 FOR X=1 TO 26
20 LET A$(X)=CHR$(X+64)
30 NEXT
40 USER VERIFY,A$
50 LET X=PEEK(64871)
60 IF X=0 THEN PRINT "Verify
    successful":STOP
70 IF X=1 THEN PRINT "Verify failed":STOP
80 IF X=2 THEN PRINT "Variable too big to verify"
```

4. Using machine-code routines

The save, load and verify routines may be called quite simply from within assembly code lines.

Saving: CALL \$F000
Loading: CALL \$F003
Verifying: CALL \$F006

Each of these routines requires one input parameter, a value in register pair DE that is the address of a list of variable names, terminated by the byte \$FF.

e.g.

```
10 CODE

8007 LD DE, VARS
800A CALL $F000 ;Save
800D RET
800E VARS: DB "A$,B$(,X,Y(,Z(,,$FF
801D RET
```

Symbols:

VARS 800E

The above machine-code routine is equivalent to the BASIC line:

```
10 USER SAVE,A$,B$(,X,Y(,Z(
```

It is a matter of personal preference whether to use machine-code calls or the USER command. The latter is probably the most straightforward.



5. Incorporating the tape routines into programs

The instructions in this section should be followed carefully and with no deviations:

(i) Load the program "VTAPE" from cassette into an MTX.

(ii) Run the program. The tape routines will be copied into high memory.

(iii) Load the program into which the tape routines are to be included. This program should be one that has been saved normally and does not auto-run. It must also not be so large that it overwrites the tape routines.

(iv) Execute the following BASIC command:

POKE 64146,255: POKE 64147,239

This changes the system variable STKLIM at #FA92, #FA93 to point to one byte below the tape routines, which has the effect of including the routines with the system variables that are saved onto tape when the program is next saved. DO NOT GO INTO PANEL TO DO THIS !!

(v) If the USER command is to be used, then the USER bytes must be patched as described above in Section 3.

(vi) The program is ready to be saved onto cassette. The variable routines and the USER bytes that point to the entry routine are now an intrinsic part of the program and will be loaded whenever the program is loaded.

Notes.

1. The USER command may do other things by patching a jump instruction to a machine-code routine at locations #F01C, #F5BD and #F5BE. This should be done after step (v).

2. The tape routines (and the program using them) can be broken into by pressing the BREAK key. If this feature is not desired, patch location #F246 with the value #18, after step (v). ★

HIGH SCORES : HIGH SCORES...Can you do better ??

GOLDMINE	4,543
ASTRO-PAC	69,390
BOUNCING BILL	14,184
SNAPPO	107,430
NUCKLES	360,200
NEMO	17,610
COBRA	8,924
MISSION ALPHATRON	49,500
TAPEWORM	86,070
TOADO	58,000+
POT HOLE PETE	39,630
MAXIMA	173,720
STAR COMMAND	140,430
PHAID	23,470
OBLOIDS	46,850
KILOPEDE	58,569
3D TACHYON FIGHTER	6,130
CONTINENTAL RAIDERS	106,240
BLOBBO	74,441

Richard Nash
Alan Dobson
Alan Dobson
Richard Franks
Sean Haverty
Richard Nash
Richard Nash
Christian Burnel
Alan Dobson
James & Vanessa (twins) Gover
Alan Dobson
B. Conway
Ian Nichols
Eric Peters
Sean Haverty
Richard Nash
Sean Haverty
Sean Haverty
David Gill

Can you beat these high scores ? Do you have a high score for a game not mentioned above ?

P.S. Joanna Gill scored over 14000 on Toado and she is only 5 years old. Embarrassing isn't it ?

Genpat's TOP SELLERS



- | | | |
|-----|------------------------|----------------------------|
| 1. | LORDS OF TIME | Level 9 |
| 2. | GRAPHICS \POTHOLE PETE | Syntax\Continental. |
| 4. | AGROVATOR | Syntax. |
| 5. | UTILITIES | Continental. |
| 6. | QOGO | Continental |
| 7. | EDASM | Syntax. |
| 8. | SNOWBALL | Level 9. |
| 9. | LITTLE DEVILS | Syntax. |
| 10. | 3D TACHYON FIGHTER | Continental. |
| 11. | MISSION OMEGA | Syntax. rising fast ! |

This chart is based on sales from within the Club only.



FOR THE BENEFIT OF NEW MEMBERS

GRAPHICS is a utility package that allows you to design Sprites, User definable, and the Ascii character set. You can then save your new designs and incorporate them within your programs without the need for Genpat statements - you treat your designs as though they were in ROM. It is a really efficient and easy way to design you characters.

EDASM: This is a really first class MACRO-ASSEMBLER. It supports all the Zilog commands including EQU & ORG which means you can load your programs into a specific memory location without the need to write re-location code. The Macro can be changed to write in 6509 6502, 6800, or your own language if you wish to do so. ** 512 only

AGROVATOR This is definitely going to be one of the best sellers of 1985. It is a completely original maze game which really does become addictive. 30 random mazes and a host of things to collect....you never get bored with this one ! ** 512 only

LITTLE DEVILS: Really good game - lots of fun and fast action. One of the favourites at Genpat.

BOUNCING BILL A very simple game and will have the rest of the family in stitches as they scoff at your downfall. Ideal fun for all the family.

HAWKWARS I had doubts about accepting this game onto the table, but the response from purchasers has made me glad I took the chance. Very fast and A b----- to play!

SOFTWARE ★ SOFTWARE ★ SOFTWARE

KEY:

Program name.....[Type of program, Issue reviewed, Price, Availability]

Type of program : BS=Business AR=Arcade TW=Tactical wargame UT=Utility

ED=Educational AD=Adventure BG=Board game CG=Card game WD=Withdrawn

Price : a=#4.95 b=#5.95 c=#6.02 d=#6.95 e=#7.95 f=#8.75 g=#12.75

h=#21.25 i=#4.50 j=#16.57 k=#13 l=#8.95

Availability : I=In stock E=Expected soon U=Unavailable

N.B. Please note that the prices quoted are special club discount prices

PAYROLL.....[BS, //, h, I]
SALES LEDGER.....[BS, //, g, U]
NEMO.....[AR, //, c, I]
SUPER MINEFIELD....[AR, //, c, I]
PHAID.....[AR, //, c, I]
TOADO.....[AR, //, c, I]
TAPEWORM.....[AR, //, c, I]
ASTROMILON.....[AR, //, c, I]
POT HOLE PETE.....[AR, 02, c, I]
MUSIC PAD.....[WD, 02, c, I]
DENNIS

& THE CHICKEN...[AR, //, c, U]
PONTOON&BLACKJACK..[CG, 06, c, I]
MAXIMA.....[AR, 01, c, I]
M CODER.....[UT, //, c, U]
JOHNNY REB.....[TW, //, c, I]
THE KEY TO TIME....[AD, //, c, I]
STAR COMMAND.....[AR, 01, d, I]
TURBO.....[AR, 02, d, I]
KNUCKLES.....[AR, //, e, I]
REVERSI.....[BG, //, e, I]
FIRST LETTERS 1....[ED, 03, f, I]
MATHS 1.....[ED, //, f, I]
SNOWBALL.....[AD, 04, f, I]
LORDS OF TIME.....[AD, //, f, I]
COLOSSAL ADVENTURE.[AD, 02, f, I]
SPELLI-COPTER.....[ED, 05, b, I]
UTILITIES 1.....[UT, //, a, I]
COMPOSER.....[UT, //, k, I]
SALTY SAM.....[AR, //, a, I]
MISSION OMEGA.....[AR, //, a, I]
BRUNWORD.....[BS, 02, j, I]
GOLDMINE.....[AR, //, c, I]
LITTLE DEVILS.....[AR, 04, a, I]
AGROVATOR.*.....[AR, 04, b, I]
HUNCHY.....[AR, //, a, I]
MISSILE COMMAND &
ARCADIANS.....[AR, //, a, I]
SON OF PETE.....[AR, //, d, E]
*FLUMMOX[AR, //, b, E]
ESCAPE FROM ZARCOS.[AR, //, d, E]
ROAD RUNNER[AR, //, a, E]
TAPE TO DISC.....[UT, //, d, I]
SPOOLER[UT, //, a, I]
BASIC EXTENSIONS...[UT, //, d, I]

PURCHASE LEDGER.....[BS, //, g, I]
BASIC BUSINESS.....[BS, //, b, I]
KILOPEDE.....[AR, //, c, I]
BLOBBO.....[AR, //, c, I]
MISSION ALPHATRON....[AR, //, c, I]
OBLOIDS.....[AR, 01, c, I]
CONTINENTAL RAIDERS....[AR, //, c, I]
ASTRO PAC.....[AR, //, c, I]
QOGO.....[AR, 02, c, I]
SNAPPO.....[AR, //, c, I]
DENNIS

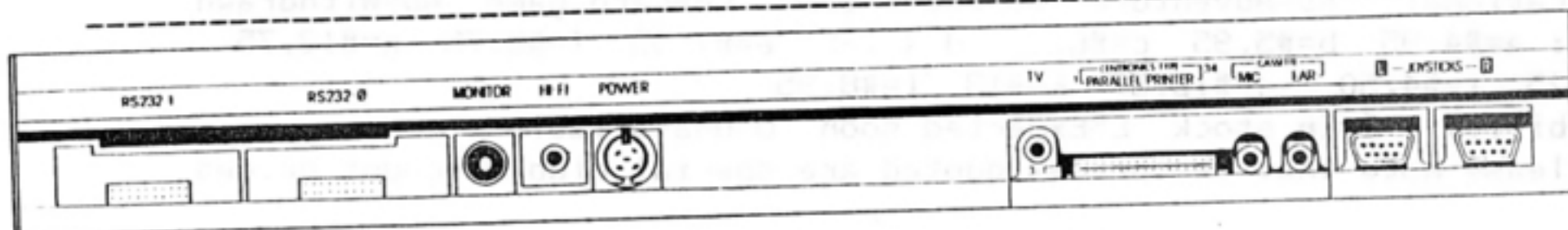
GOES BANANAS.....[AR, //, c, U]
THE ZOO GAME.....[AD, 03, c, I]
GAUNTLET.....[AR, //, c, U]
COBRA.....[AR, //, c, I]
MURDER AT THE MANOR....[AD, //, c, I]
FIREHOUSE FREDDIE.....[AR, 06, c, I]
DRAUGHTS.....[BG, //, d, I]
3D TACHYON FIGHTER....[AR, 04, d, I]
BACKGAMMON.....[BG, //, e, I]
CHESS.....[BG, //, f, I]
WORD & PICTURE.....[ED, 04, f, I]
PHYSICS 1.....[ED, //, f, I]
ADVENTURE QUEST.....[AD, //, f, I]
DUNGEON ADVENTURE.....[AD, 01, f, I]
RETURN TO EDEN.....[AD, //, f, I]
HELI-MATHS.....[ED, 05, b, I]
TUMBLEDOWN TOWER.....[AD, //, i, I]
EDASM.*.....[UT, //, e, I]
DOODLEBUG DESTROYER....[AR, //, a, I]
GRAPHICS.....[UT, 01, b, I]
THE MAN FROM GRANNY.*..[AD, //, a, I]
ALICE IN WONDERLAND....[AD, //, c, I]
HAWKWARS.....[AR, //, a, I]
BOUNCING BILL.....[AR, 06, a, I]
BRIDGE.*.....[CG, //, d, I]
MEMWORD.....[BS, //, l, E]
MEMOSKETCH.....[UT, //, b, E]
QOGO 2[AR, //, d, E]
CANVAS[UT, //, d, E]
CHAMBEROIDS[AR, //, d, E]
LES FLICS[AR, //, d, E]
USER EXTEND[UT, //, e, I]
H & L DUMP[UT, //, a, I]
DATA FILE[UT, //, g, I]

* runs only on the MTX512 and RS128

TAPE OF THE MAG #3.50 inc p & p

DATA ANALYSIS : MOUSE : HEX DUMP : DOG FIGHT : DATA SAVE AND LOAD

Please note that we have decided to include the Data Save & Load within the Tape of the Mag, so please ignore earlier comments.



To start, I would like to welcome our new members from Astralia & New Zealand.

The Club's lable Syntaxsoft has really caught on - we have had telephone calls from all over the world and the lable will go on sale in those countries that have asked for marketing rights. New software continues to flow into Genpat and the end of the month should see most titles released. The Club is negotiating with Memotech for the rights to market **Dennis & the Chicken** and **Dennis Goes Bananas**. **Meggastar** will release the long awaited **Son of Pete & Qogo 2** by the end of this month, and the Club expects to be the first to hold stocks of these two titles. **Sentient** are almost ready to release two utilities (one reviewed earlier), we shall bring you details in March. **Syntaxsoft** will release what is definitely one of the most unique games of this year - written by Paul [Agrovator] Johnson - called **Flummox**, by the end of the month. We also have a new game called **Iceburg** which will suit all those **Astroid** fans.

The Club has been negotiating with **Tri - to get it to you sooner or later** - **Com** (as one member has put it!), and if Francis Wallinger keeps his word, we will take over **all the titles**. The Club will then be willing to supply all members, who have written to say they have not received their orders, with the software in question at 50p per title which will cover our postage costs, providing, as I have said, **Tri-Com** keep their word! The titles will then be put on general release under the Syntax lable.

We have almost caught up with the back-log and any member awaiting a reply from me personally, should hear from me within the next few days. Members waiting for Back Issues please be patient, we have had to do a re-print of all editions.

Please do write to the magazines and if you get any interesting replies, send them to us, and we can publish them. We have now started work on the March Edition, and I can promise you some very interesting programs next month.

Finally, here is a little puzzle sent to us by Geoff Boyd, Memotech's chief designer: **IF NOT (222) AND (323) AND (727) THEN (223) AND (323) AND (727) ELSE CENTRONICS. COLLECTORS EDITION OF QOGO 2 FOR FIRST CORRECT ENTRY.**