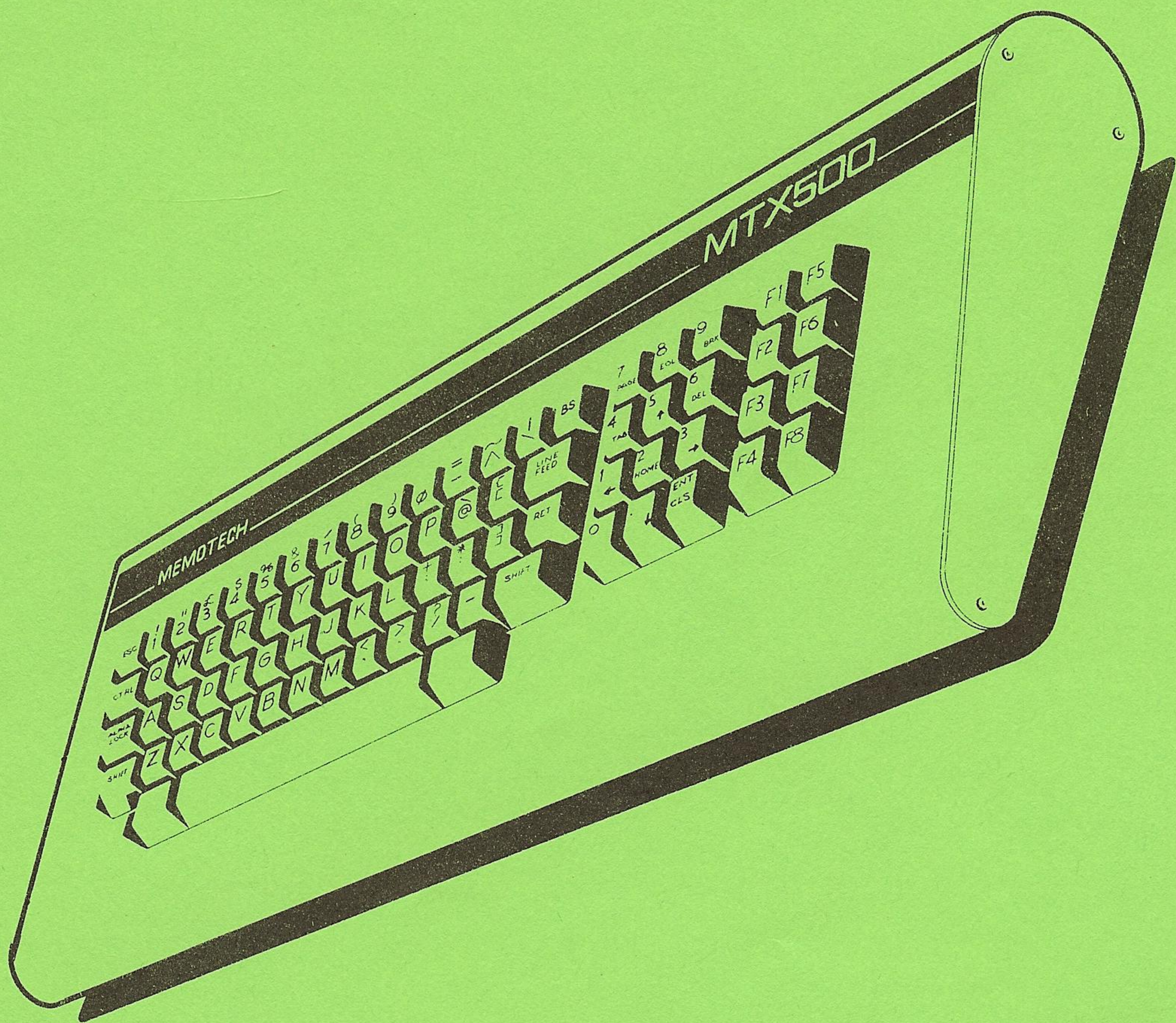


memorad

Memotech Computer User Club Magazine





EDITORIAL

Many thanks to you all for complying with my request not to telephone on Wednesday and Friday during the day. I really appreciate your consideration.

If you have written in with a request for technical information and have not yet received a reply, please do not worry. The past two weeks have been devoted to getting all the software orders out to members in time for Christmas. I will try to answer all letters before the festivities start.

Genpat will be closed from 6-00pm December 23rd until Monday 7th January 1985. However, even though I am on vacation, any member who has an urgent problem can leave a message on the answerphone, and I will ring each evening to listen to the messages.

1985 will see many exciting ideas brought to fruition. Jim Wills at Memotech has assured me that quite a number of new companies will be writing software for the MTX. The Club's own table will have at least 10 new titles available before the end of February.

I would like to welcome all the new members - and there have been many - who have joined during the past three weeks, and I hope they will soon get into the swing and start sending in those programs. At this point I would like to reiterate a request made in an earlier edition: **We need articles on hardware interfacing, lots more programs, CP/m and business software listings, and anything else you may think will be of interest to other members.**

The Club will have the first **Graphic Adventure** available within the early part of the new year. We have also signed **Bouncing Bill** by the author of **The Zoo**. **Bouncing Bill** is a smashing game. Very simple in its concept, but absolutely hilarious to play fun for all the family.

We are about to start a special section on **Education & the MTX** so if there are any teachers who would like to contribute, we would like to hear from you.

Finally, many thanks for all those nice Christmas cards, and I would like to take this opportunity to wish you all **A MERRY CHRISTMAS and A HAPPY NEW YEAR.**

K.



This magazine is the copyright of Genpat. No part may be reproduced without the prior consent of Genpat. All programs included within the magazine are for personal use only, and may not be sold for gain or profit.

Printed by Genpat 3, Bulcock Street, Burnley, BB10 1UH 0282-57427

Programming in Pascal

```

PROGRAM GraphicsAndSpriteExtensions;
(   MTX graphic extension package   )
(   Vers 2 S.Varley MEMBRAIN Software )
PROCEDURE RST10(N:1..12;DATA:ARRAY[1..12] OF CHAR);
VAR I:INTEGER;
BEGIN
POKE(£C000,CHR(£DD));POKE(£C001,CHR(£E5));POKE(£C002,CHR(£D7));
DATA[1]:=CHR(ORD(DATA[1])+£80);
FOR I:=1 TO N DO
POKE(£C002+I,DATA[I]);
POKE(£C002+I,CHR(£DD));POKE(£C003+I,CHR(£E1));POKE(£C004+I,CHR(201));
USER(£C000)
END;

```

```

PROCEDURE view(dir:0..7;dis:0..255);
VAR n:1..12;data:ARRAY[1..12] OF char;
BEGIN
n:=4;data[1]:=chr(3);data[2]:=chr(20);data[3]:=chr(dir);data[4]:=chr(dis);
rst10(n,data)
END;

```

```

PROCEDURE mvspr(p:0..15;sn:1..32;d:integer);
VAR n:1..12;data:ARRAY[1..12] OF char;
BEGIN
n:=5;data[1]:=chr(4);data[2]:=chr(19);data[3]:=chr(p);data[4]:=chr(sn);data[5]:=chr(d);
rst10(n,data)
END;

```

```

PROCEDURE adjspr(p:0..5;sn:1..32;v:0..255);
VAR n:1..12;data:ARRAY[1..12] OF char;
BEGIN
n:=5;data[1]:=chr(4);data[2]:=chr(17);data[3]:=chr(p);data[4]:=chr(sn);data[5]:=chr(v);
rst10(n,data)
END;

```

```

PROCEDURE ctlspr(p:0..6;sn:0..255);
VAR n:1..12;data:ARRAY[1..12] OF char;
BEGIN
n:=4;
data[1]:=chr(3);data[2]:=chr(14);data[3]:=chr(p);data[4]:=chr(sn);
rst10(n,data)
END;

```

```

PROCEDURE sprite(sn:1..32;p:0..127;xp,yp:-4095..4095;xs,ys:-128..127;col:0..15);
VAR n:1..12;data:ARRAY[1..12] OF char;
BEGIN
n:=11;data[1]:=chr(10);data[2]:=chr(18);data[3]:=chr(sn);data[4]:=chr(p);data[5]:=chr(xp MOD 256);data[6]:=chr(xp DIV 256);
data[7]:=chr(yp MOD 256);data[8]:=chr(yp DIV 256);data[9]:=chr(xs);data[10]:=chr(ys);data[11]:=chr(col);
rst10(n,data)
END;

```

```

PROCEDURE csr(x:0..39;y:0..23);
VAR N:1..12; DATA:ARRAY[1..12] OF CHAR;
BEGIN
N:=4;DATA[1]:=CHR(3);DATA[2]:=CHR(3);DATA[3]:=CHR(X);DATA[4]:=CHR(Y);
RST10(N,DATA)
END;

```

```

PROCEDURE COLOUR(P:0..4;N:0..15);
VAR C:1..12; DATA:ARRAY[1..12] OF CHAR;
BEGIN

```

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

```

C:=4;DATA[1]:=CHR(3);DATA[2]:=CHR(16);DATA[3]:=CHR(P);DATA[4]:=CHR(N);
RST10(C,DATA)
END;

PROCEDURE GENPAT(P:0..7;N:0..154;D1,D2,D3,D4,D5,D6,D7,D8:0..255);
VAR C:INTEGER; DATA:ARRAY[1..12] OF CHAR;
BEGIN
C:=12;DATA[1]:=CHR(11);DATA[2]:=CHR(15);DATA[3]:=CHR(P);DATA[4]:=CHR(N);DATA[5]:=CHR(D1);DATA[6]:=CHR(D2);DATA[7]:=CHR(D3);DATA[8]:=CHR(D4);
DATA[10]:=CHR(D6);DATA[11]:=CHR(D7);DATA[12]:=CHR(D8);
RST10(C,DATA)
DATA[9]:=CHR(D5);
END;

PROCEDURE ATTR(P:0..3;STATE:0..1);
VAR C:INTEGER; DATA:ARRAY[1..12] OF CHAR;
BEGIN
C:=5;DATA[1]:=CHR(4);DATA[2]:=CHR(27);DATA[3]:=CHR(65);DATA[4]:=CHR(P);DATA[5]:=CHR(STATE);
RST10(C,DATA)
END;

PROCEDURE GR(X:0..255;Y:0..191;B:1..8;VAR CH:CHAR);
VAR C:INTEGER; DATA:ARRAY[1..12] OF CHAR;
BEGIN
C:=6;DATA[1]:=CHR(5);DATA[2]:=CHR(27);DATA[3]:=CHR(67);DATA[4]:=CHR(X);DATA[5]:=CHR(Y);DATA[6]:=CHR(B);
RST10(C,DATA);
CH:=PEEK($FE1A,CHAR)
END;

BEGIN
vs(4);page;
genpat(3,1,255,199,199,199,8,1,199,255);
ctlspr(0,1);ctlspr(2,1);ctlspr(5,1);ctlspr(6,1);
sprite(1,1,100,100,1,1,15);
REPEAT UNTIL (inch='S') OR (inch='s')
END.

```

3D TACHYON FIGHTER by ANGUS GRANDISON

This piece of software is loosely based on the great arcade game **Buck Rogers**. For those of you who have not seen this game it is basically a 3D shoot 'em up. You must destroy a certain number of aliens before you can advance to the next screen, which is harder and more difficult.

The action does not follow **Buck Rogers** exactly but the game is still very good. The game is well thought out and is fast. The sound is alright and the graphics are great, except for the 3D tunnel which is cronic. After you have gone through 4 different screens you start again but this time it's harder.

My only criticism is that it is nearly impossible to play this game without a joystick because of the way the keypad is configured. On the whole I loved this game and would recommend it to anybody.

AGROVATOR by TIM ROTHWELL

This game at first sight looks like yet another version of that old arcade favourite Pacman. However, this one has a number of new features which make you wish the original had been written in this form. The screen you are presented with on completion of loading gives a list of the 16 different 'foods' (ranging from cherries worth 100 points to mutants worth 1600 points) you can eat whilst munching your way through the dots. A press of the fire button and you are into the first screen.

The playing screen gives information on your current score, high score, number of lives (originally 3), number of bullets (originally 3), frame (or screen) number and the number of dots left to be eaten. The provision of fire power gives you the opportunity to get out of those tight spots by pressing the fire button and releasing a bullet behind you. Apart from the fact that you only have 3 bullets there is one more restriction, you can only fire while moving. Two more bullets can be obtained by eating a 'changer' which appears at a random position every now and then. You can eat the 'swirlers' (or ghosts) after eating one of the clocks, but you have only 8 to 10 seconds to get them before they change back.

The maze drawn is selected at random from at least 29 different ones (I never saw a screen with a number higher than that). The variety in the design of the screens means that you don't get bored with the same or similar mazes. The sound is simple but very effective and doesn't grate as some sound effects do. The game is excellent value at £5.95 and will probably be a best seller through Christmas and into the New Year (if it isn't then it's your loss). It is a program which will make those MTX500 owners wish they had spent the extra money and bought the MTX512.

LITTLE DEVILS by TIM ROTHWELL

This program, written by Mark Lawrence, is an original one as far as I know and the programmer should be congratulated for that. It's a great game which offers 8 levels of play with a bonus after every 8 levels. The screen consists of a few simple walls, ice cubes, bells, four little devils (apt name) and you. The aim of the game is to destroy the little devils before they get you. To achieve this you throw ice cubes at them or after ringing a bell (causing the little devils to turn into balloons) by jumping on them. The little devils move very quickly so you have to be a bit useful with the joystick or keys.

The programmer has thoughtfully provided a hold facility (for those who tire easily), a named high score feature and a very interesting loading sequence. At £4.95 this program is excellent value and I hope that Syntax Software sell a large number of copies as Mark Lawrence deserves to be successful with this game. I believe this is another indication that the quality of software for the Memotech computers is improving.

PASCAL UPDATE by Bina Bhuptani

The "include file" option of the Pascal compiler will not work in the following format :-

```
Program Hello;
var a : char;
{$F FILENAME}
BEGIN
writeln ("Hello")
END.
```

but will work in this format :-

```
Program Hello;
var a : char;
    b : char;
{$F FILENAME}
BEGIN
writeln ("Hello")
END.
```

The option has to begin on an even address.

SNOWBALL by COLIN REES

When the editor told me that I was to be given the opportunity to review an adventure game from Level 9, I little realised how much time and how many late nights would be consumed by the task. **Snowball** is a mammoth adventure game in which you play the part of a secret agent called Kim Kimberley. Your task is to save **Snowball 9**, a five mile long spaceship, with one million colonist aboard.

In **Snowball** you are presented with excellent descriptive texts throughout the game. Read this - "You are in a large low room surrounded by lights and vids. A barely audible high pitched humming emerges from a vast display panel with the legend "SNOWBALL IX FREEZER CONTROL". To the south a luminous cyladder rises through the ceiling. Exits lead North, East, South and Up."

After many hours and a host of hints from the editor I still only managed to clock up a score of 400 out of a possible 1000, and I could not bring myself to switch the micro off. It is a very addictive game which has absorbed my attention for many hours at a stretch. Luckily a 'Save game' feature is included in the program. This is highly recommended.

GRAPHICS

*The BEST thing since
sliced bread*

*Take the sweat out of
designing your own, and
exclusive graphics or
characters.....*

*Available only
From GENPAT*

£5.95 inc P.P



MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

CIRCLE by CLIVE TAYLOR

This is a short routine which will draw apparently true circles on the screen. It uses the principle of plotting an ellipse with the y-axis being longer to counteract the horizontal stretching of the pixels. The ratio between the x and y axes is 3:4.

```
10 VS 4:CLS
20 FOR A=1 TO 360
30 LET B=A*(2*(PI/360)) : REM Converts degrees to radians
40 LET X=100+30*COS(B)
50 LET Y=100+40*SIN(B)
60 PLOT X,Y
70 NEXT A
80 GOTO 80
```

In lines 40 and 50, the first parameter after the = sign are the x and y co-ordinates of the centre of the ellipse. Provided the next parameter (i.e. the 30 in line 40 and the 40 in line 50) are kept in the ratio 3:4 the ellipse produced will appear circular on the screen.

HACKERS TIP by STEPHEN WHITE

By using the following information you can obtain extra lives in Snappo. Just a word of warning - if you choose to make this alteration it does tend to take a lot of fun out of the game because there is no challenge left.

Follow the instructions below depending upon your machine.

LOAD "SNAPPO", then RESET the machine and type.

MTX500

```
POKE 40861,no. of lives (0-255)
POKE 64164,88
POKE 64165,164
POKE 64167,88
POKE 64168,164
POKE 64172,88
POKE 64173,164
VS 4: CLS: RUN
```

MTX512

```
POKE 24477,no. of lives (0-255)
POKE 64164,88
POKE 64165,64
POKE 64167,88
POKE 64168,164
POKE 64172,88
POKE 64173,64
VS 4: CLS: RUN
```

To save the program with the above alteration, type GOTO 500 instead of VS 4: CLS: RUN'.

USING THE MTX FRONT PANEL

The Front Panel is a debugging tool which will allow the machine code programmer to look at or single step through an assembly program. Most assembly programs move data between registers, store data in memory locations, or carry out some form of test on the **Flag** register. Program failure often occurs due to one of these operations performing in an unexpected way. In the days before **assemblers** one had to debug a program by single stepping through a paper listing and playing computers on paper. Assemblers and monitors [Front Panel] have made this task far easier.

To use the Panel you must first become familiar with the type of operations that can be performed from the keyboard. They are as follows:

KEY	OPERATION			
RETURN	MOVES	MEMORY	CURSOR	FORWARD
DOWN ARROW	"	"	"	DOWN
UP ARROW	"	"	"	UP
- [MINUS SIGN]	"	"	"	BACKWARD
. [FULL STOP]	"	REGISTER	"	

TYPING

LIST will list from current memory location and typing LIST £4000 will list from £4000.

TYPING

DISPLAY will display current location of Memory Block cursor. Typing DISPLAY £4C00 will display from £4C00.

There are two cursors which are displayed as '>'. One is the **Memory** cursor, and one the **Register** cursor.

Typing I toggles the memory display between a hexadecimal display and a Ascii display.

How to use the front panel is best explained by example. Before we continue type in the following listing..... it doesn't do much, but will serve as a demonstration program. First enter the assembler by typing **ASSEM 10 <RET>** in answer to the **Assemble>** prompt press <RET>. Your display should now look like this:

```
8007      RET      [MTX 500]      or      4007      RET      [MTX 512]
```

If any of the code does not overwrite the RET you must erase the remains by pressing EOL key. Now press <RET> and your display should be as follows:

```
800B      RET      [MTX 500]      400B      RET      [MTX 512]
```

Now type in the rest of the code.

USING THE FRONT PANEL TO LOOK AT THE LISTING

After saving the program to tape enter the front panel by typing PANEL <RET>. Your display should now look something like this:.....

10 CODE		401F LOOP: JR LOOP		AF >0000 F3
		4021 ADD: ADD A,L		BC 0000 F3
4007	LD SP,(EFA96)	4022	ADD A,H	DE 0000 F3
400B	LD A,E10	4023	RET	HL 0000 F3
400D	CP E40	4024	RET	IX 0000 F3
400F	CP E10			IY 0000 F3
4011	CP E5	Symbols:		SP 0000 F3
4013	LD L,A	ADD 4021 LOOP 401F	DI	PC 0000 F3
4014	LD H,E3C			
4016	LD DE,E4500			FFF0: 80 07 53 0A 11 1C 53 0A
4019	EX DE,HL			FFF8: 00 00 00 00 00 00 00 00
401A	PUSH HL			0000: >F3 AF 21 00 40 C3 94 01
401B	POP BC			0008: 5E 23 56 23 C9 FF FF FF
401C	CALL ADD			0010: E3 F5 7E FE 40 C3 FA 06
				0018: C3 74 3B D7 2D 0A C9 00

Now type LIST 4007 <RET> MTX 500 owners type LIST 8007 <RET>. Your video display will now look like the following dump.

4007	LD SP,(EFA96)	4007	LD SP,(EFA96)	
400B	LD A,E10	400B	LD A,E10	AF 0000 F3 ←
400D	CP E40	400D	CP E40	BC 0000 F3
400F	CP E10	400F	CP E10	DE 0000 F3
4011	CP E5	4011	CP E5	HL 0000 F3
4013	LD L,A	4013	LD L,A	IX 0000 F3
4014	LD H,E3C	4014	LD H,E3C	IY 0000 F3
4016	LD DE,E4500	4016	LD DE,E4500	SP 0000 F3
4019	EX DE,HL	4019	EX DE,HL	PC >4007 ED
401A	PUSH HL	401A	PUSH HL	
401B	POP BC	401B	POP BC	
401C	CALL ADD	401C	CALL ADD	
401F LOOP: JR LOOP		401F LOOP: JR LOOP		
4021 ADD: ADD A,L		4021 ADD: ADD A,L		

THIS REPRESENTS WHAT IS IN MEMORY LOCATION POINTED TO BY REGISTERS.

REG CURSOR

DI		LD SP,(EFA96) FIRST INSTRUCTION
3FF0: F2 CB F6 ED 42 CD A0 3F		3FF0: F2 CB F6 ED 42 CD A0 3F
3FF8: D1 C5 CD A0 3F D1 18 E4		3FF8: D1 C5 CD A0 3F D1 18 E4
4000: 7B 00 0A 00 C2 1E 00 ED		4000: 7B 00 0A 00 C2 1E 00 ED
4008: 7B 96 FA 3E 10 FE 40 FE		4008: 7B 96 FA 3E 10 FE 40 FE
4010: 10 FE 05 6F 26 3C 11 00		4010: 10 FE 05 6F 26 3C 11 00
4018: 45 EB E5 C1 CD 21 40 18		4018: 45 EB E5 C1 CD 21 40 18

MEMORY CURSOR.

You can see that your program is listed on the left side of the screen, and if you were to press LIST <RET> again the display would move to the next part of the listing, and by continuously typing LIST you can move through a complete program. Try it later with TOADO.... press reset and then enter the panel and LIST from 4007 or 8007 and you will see that the code for TOADO is still there even though you have reset the machine !

Anyway, back to our program. Now type 'D 4007' from now on 500 owners assume that the '4' means '8'....<RET>. You will now see that your program is listed at the bottom of the screen as a Hexidecimal Dump

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

with the memory cursor '>' pointing to location 4007. To get out of this mode press <BRK> key. Pressing the UP or Down arrows will move this cursor either one byte up or one byte down memory.

To single step through the program you must first set the PC [Program Counter] to the start of the program, or to the memory location you wish to single step from. To do this press '.' [full stop]. You will see that the '>' register cursor moves down the list of registers in the top right hand corner of the screen. Stop when the cursor points at PC. Now press 'R' in answer to the REGISTER > prompt type 4007 <RET>. You will now see that the PC = 4007, and below the registers the instruction LD SP(&FA96) is displayed. If you now press 'S' you can see that the latter instruction is changed to LD A,&10. There will now be a value displayed in the AF register. Continue pressing 'S' and you will locate the next instruction CP &40. The next time you press 'S' you will see that the Flag Status Bits which are displayed above the registers begin to display information. Because the value in the A register is less than the compared value [40] the 'Carry Flag' [C] is displayed to denote a CARRY. Continue pressing 'S'. When compared with &10 the Zero Flag [Z] will be displayed, and when compared with &5 the Carry Flag [C] is not displayed because there is no CARRY.

As you continue pressing the 'S' key notice what happens to the registers in relation to your code listing, and you will see that each operation is carried out and reflected in the registers. Pay particular attention to the the Stack Pointer [SP] when you get to the PUSH, POP, and the CALL instructions. You should be aware that as a PUSH is performed the SP moves two places down in memory, and the POP moves the SP up two places. The same is true for the CALL & RET instructions.

At any point you can move the register cursor > to another register, press R and change the value in that particular register..... go on try it.... you can't do any harm ! In this way you can modify your program as you find a mistake or wish to try a different value.

The T key performs in exactly the same way as the S key except that it treats CALLS in a different manner. Whenever you reach a CALL instruction using the S key, the panel then single steps through the subroutine. If you use the T key the program registers the call, but will carry it out without single stepping through the instructions. Set the PC to 4007 then single step using the S key, then try the same thing using the T key. Notice the difference ?

Another useful feature of the front panel is the Go & To commands. These commands are accessed by pressing 'G'. The answer to the Go prompt specifies the start, To requests the finish address. These two commands allow you to test a section of code under run-time conditions and then check the registers for the expected values..... this saves having to single step if you are only interested in the end result. For example : pressing G and responding to the prompt with 4007 and answering the TO prompt with 801F will result in the Panel running through the assembly code.

M command allows the movement of code from one address to another. The parameters are Move <start address> <RET> <end address> <RET> to

Location <new address><RET>. This command is useful for moving blocks of code to new locations within Ram. Try this:-

Type M in answer to Move> type 0000 <RET>. When the End> prompt appears type 1000<RET>. Answer the To> prompt with 4007<RET>. If you now type L 4007 <RET> you will see that the ROM from location 0000 has been moved to 4007 in your RAM. Well; haven't you always wanted to look through ROM ?

Entering a B will result in the panel asking if you want to return to Basic. Typing Y will return you there.

A word of caution.... after pressing the D for display anything you type in will be transferred to memory as soon as you hit the <RET> key. Very useful. But how do you get out of this mode ? Hit the <BRK> key !

Here is a section of code that does do something useful....it allows you by pressing P to dump the Front Panel to a printer. Pressing F1 will clear all the registers for you..... Thanks to Membrain Software.

```
10 REM Panel Extensions
20 REM P print Front Panel
30 REM F1 put 0 in all registers
31 REM S.Varley MEMBRAIN Software.
40 CODE
```

```
407A      LD DE,£F6F0
407D      LD (£FA92),DE
4081      LD (£FA9F),DE
4085      LD HL,START
4088      LD BC,300
408B      LDIR
408D      LD A,£C3
408F      LD (£FA9E),A
4092      RET
4093 START: LD A, (£FD7D)
4096      CP 128
4098      JR NZ,PRINT
409A      RST 10
409B      DB £93,"Registers cleared.."
40AF      LD B,200
40B1 LP:   HALT
40B2      DJNZ LP
40B4 ZEROREG: LD DE,£FD91
40B7      LD B,16
40B9      DI
40BA ZERO: LD A,0
40BC      LD (DE),A
40BD      INC DE
40BE      DJNZ ZERO
40C0      EI
40C1      RET
```

```
40C2 PRINT: LD A, (£FD7D)
40C5      CP "P"
40C7      RET NZ
40C8      RST 10
40C9      DB £8C,"Printing...."
40D6      RST 10
40D7      DB £84,27,"XDA"
40DC      DI
40DD      LD B,27
40DF      CALL £0CE3
40E2      LD B,"Q"
40E4      CALL £0CE3
40E7      LD B,40
40E9      CALL £0CE3
40EC      LD A,0
40EE      OUT (2),A
40F0      LD A,28
40F2      OUT (2),A
40F4      LD DE,960
40F7 NEXT: IN A,(1)
40F9      LD B,A
40FA      CALL £0CE3
40FD      DEC DE
40FE      LD A,D
40FF      OR E
4100      JR NZ,NEXT
4102      EI
4103      RST 10
4104      DB £84,27,"XDE"
4109      RET
```


MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

JOHN MULLINS SHOWS YOU HOW TO UTILISE MTX INTERRUPTS

One question I am frequently asked about the MTX concerns interrupts, and how they can be handled by the programmer. With the MTX there are many ways that this can be done, but for this month I will concentrate totally on the facilities offered by the machine operating system, and next month I will move onto the more complicated topics of the Z80 interrupt modes, the I register and the operation of the CTC.

Every 1/125th of a second the MTX is interrupted and the Z80 CPU halts execution of it's main program, (usually the large machine code program we all know and love (or hate!) as BASIC), and the interrupt handling routine is entered. The interrupt handler first toggles bit 7 of INTFFF (£FD5E) and then updates the real time clock. Once this has been done bit 7 of INTFFF is tested and if set the interrupt handler is exited, however if it is reset the bits which correspond to cursor flash, break key testing etc. are tested and if set the appropriate routines are called. Once this has been performed the bits which correspond to USER defined interrupt service routines are tested, and if set a call to USERINT (£FA98) is performed. Thus all that is required to activate a user defined interrupt routine is to point USERINT at it with a jump instruction and set one of the USER bits of INTFFF. An important point to note is that if more than one user bit is set the user routine will be called more than once.

Because interrupts stop execution of one program whilst another is executed the only way execution of the main program can resume correctly is if absolutely nothing is corrupted by the interrupt handler. This means that any registers used by the interrupt handler must be restored to the correct values before returning, and more importantly the state of the keyboard and the VDP must not be altered. Fortunately the author of the ROM also realised this could pose a problem and took various steps to ensure that a conflict never occurs. In order to read the break key, which is done during the interrupt handler, the state of the keyboard will be altered, so in order to ensure that any value output onto the keyboards drive lines is not corrupted the value is stored at LASTDR (£FD7E) immediately before the byte is sent to the keyboard. On exit from the interrupt routine the value held in LASTDR is output to the drive lines thus ensuring keypresses are detected correctly. Similarly when the user is writing to the screen a non-zero value should be placed in VINTFG (£FF58) to indicate that it is unsafe to write to the screen during the interrupt. These are important points to remember when writing machine code programs which run with interrupts enabled.

Now with all the technicalities out of the way we will develop a short interrupt driven routine which displays the real-time clock in the top left-hand corner of the text screen and can be toggled on and off by pressing the F1 key to turn it on and F4 to turn it off. Since this routine will write to the screen and read the keyboard it provides a nice way of illustrating the above points. First type in the listing as it is printed (omit the comments if desired!), then save to tape or disc then as direct commands enter the following:-

MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

POKE 64154,128 (for MTX 500) or 64 (MTX 512)
POKE 64153,7
POKE 64152,195

This basically points USERINT to the start of our servicing routine. It is good practice when entering jumps to do it in reverse order as above, otherwise if the JP instruction is entered first a jump to some non-descript address may occur and all sorts of problems arise (usually a total system crash!!). Now all that remains to do is activate the servicing routine thus:

POKE 64862,131

Now press F1 and the clock appears and ticks away merrily, press F4 and it stops, and disappears when something is printed over it's old position. Try omitting the checks as to whether it is safe to write to the screen, and leave out the part which outputs the LASTDR onto the keyboard and see just what problems arise. Finally, a point worth mentioning is that this routine will only work on the text screen due to the graphics screen being used as a bit mapped display (see last months Memopad), but you now have enough information to put this into practice so go on give it a whirl the only way to learn is by doing! Have fun, I'll be back next month with more exciting ways to use interrupts see you then.

10 CODE

8007	PUSH AF ;Save used registers	803A	INC HL ;by a colon for hours and minutes
8008	PUSH BC	803B	OUT (1),A
8009	PUSH HL	803D	LD A,(HL)
800A	LD A,\$FE ;First test if F1 is being pressed	803E	INC HL
800C	OUT (5),A	803F	OUT (1),A
800E	IN A,(6)	8041	LD A,1 ;If printing seconds do not print
8010	BIT 1,A	8043	CP B ;a colon
8012	CALL Z,CLOCKON ;Set flag to indicate clock on	8044	JR Z,EXIT
8015	LD A,\$7F ;Now test if F4 is being pressed	8046	LD A,\$3A ;\$3A = ":"
8017	OUT (5),A	8048	OUT (1),A
8019	IN A,(6)	804A	DJNZ LOOP
801B	BIT 1,A	804C EXIT:	LD A,(\$FD7E) ;Get last drive and place on keyboard
801D	CALL Z,CLOCKOFF ;Set flag to indicate clock off	804F	OUT (5),A ;drive lines
8020	LD A,(FLAG) ;Is clock on ?	8051	POP HL ;Retreive used registers
8023	OR A	8052	POP BC
8024	JR Z,EXIT ;Exit routine if not	8053	POP AF ; and return from routine
8026	LD A,(\$FF58) ;Is it safe to write to the screen?	8054	RET
8029	OR A	8055 CLOCKON:	LD A,1
802A	JR NZ,EXIT ;Exit if not	8057	LD (FLAG),A
802C	LD A,1 ;Set up VRAM address \$1C01	805A	RET
802E	OUT (2),A ;Equivalent to CSR 0,0 in VS 5	805B CLOCKOFF:	XOR A
8030	LD A,\$5C	805C	LD (FLAG),A
8032	OUT (2),A	805F	RET
8034	LD HL,\$FD57 ;Address of real time clock	8060 FLAG:	DB 0
8037	LD B,3 ;Used as a counter	8061	RET
8039 LOOP:	LD A,(HL) ;Print two characters followed		

Symbols:

FLAG	8060	EXIT	804C
LOOP	8039	CLOCKON	8055
CLOCKOFF			805B

 MOUSE ROUTINE by ADRIAN MORRIS

You asked for a program - well here it is. It's a mouse simulation which uses the cursor control keys or joystick to move an arrow up and down a menu. When the arrow points at the desired option pressing the <HOME> key will select it. I didn't put any colour in because I've only got a black and white telly, so it can be improved in that respect. For kids a man going up and down a ladder might justify the circuituous route to option selection.

```

10 VS 4: CLS
20 CSR 5,5: PRINT "1. OPTION ONE"
30 CSR 5,7: PRINT "2. OPTION TWO"
40 CSR 5,9: PRINT "3. OPTION THREE"
50 CTLSPR 0,1: CTLSPR 1,1
60 CTLSPR 2,1: CTLSPR 6,0
70 LET P=148
80 GENPAT 3,0,24,4,2,255,255,2,4,24
90 SPRITE 1,0,32,P,0,0,1
100 ADJSPR 3,1,P
110 LET IN$=INKEY$
120 IF IN$="" THEN GOTO 110
130 IF ASC(IN$)=26 THEN GOTO 110
140 LET P=P-(ASC(IN$)=11)+(ASC(IN$)=10)
150 IF P<116 THEN LET P=116
160 IF P>148 THEN LET P=148
170 GOTO 100
180 LET D=INT(192-P)/8
190 CSR 5,0
200 LET X=VAL(SPK$)
210 IF X=0 THEN GOTO 110
220 CLS
230 REM ON X GOTO
  
```

 LITTLE DEVILS
 AND THEY ARE

*This is a unique game
available only on the
MTX series.....*

SYNTAX SOFTWARE

£4.95 inclusive

*A game for the
whole family.....*



 PRINTER UPDATE by W.T. CRASWELL

Here are the changes to Richard Sargent's printer program to use with the DMXB0 printers.

```

34 LPRINT CHR$(27); "3"; CHR$(24);
40 LPRINT CHR$(27); "K"; CHR$(0); CHR$(2);
63 LPRINT CHR$(27); "K"; CHR$(0); CHR$(2);
83 LPRINT CHR$(27); "@"; CHR$(13);

ADD -- 31 LPRINT CHR$(27); "P"; CHR$(0);
  
```

WORD AND PICTURE Continental Software

The program starts by allowing the 'teacher' to change a number of different options including the number of questions asked, the number of wrong answers allowed and the word list. The program then displays a pictorial representation of the word and as the bar moves down the option list the child has to choose the word to match the picture. After each word a little man gets in on the act but I will leave you to discover his role.

An evaluation of Continental's educational programs for young children by Mr. B. Cooke.

This program shares many of the features already found in 'FIRST LETTERS' (see Issue 3 for that review). The progression to multiple choice answers is seen as excellent preparation for any ongoing studies.

The material contained in the two programs tested gives early familiarisation with the home computer as a teaching/learning medium. The need for correct spelling is soon realised and the animated reward screens tend to remove some of the formality from learning.

CLANGERS !!!!!

A couple of important typing errors crept in on page 26 of Issue 3. I'm sorry to report that the 100K drive costs £235 (not £135) with the RS232 interface and the 250K drive costs £249 (not £149) with the RS232 thrown in free. Genpat tries its best to get you the best value in hardware and software but I'm afraid we can't miracles (yet!).

MRY HD LITTLE LMB
ITS FLEESE WS WHITE
S SNOW



ND EVERY WERE
TT MRY WENT TE
LMB WS SURE TO GO...



HMM, FIRST THE "H",
NOW THE "A"....



ASSEMBLY LANGUAGE PART 4

No matter what type of program you are writing, be it an arcade game, word processor, or screen display, much of the program will involve moving data either from one memory location to another or from the CPU registers to memory. Most of the time the CPU registers will be involved in moving data or temporarily storing it.

One of the most important areas in memory is the **stack area**, where the CPU stores data temporarily and then retrieve the values when needed. Understanding how the stack operates is most important when using assembly language: **if the stack gets overwritten or destroyed, your program will crash !** The stack can reside anywhere in memory, and setting the location involves loading the **stack pointer** with the correct memory address. Obviously, this address must be in **RAM** memory, and is a simple operation: **LD SP,£4007**. Care must be exercised when choosing a location, and you must always be aware that the **stack builds down from high addresses to low addresses**. You must also make sure that your program cannot overwrite you stack area. On the MTX it is always wise to load the stack from **system variable £FA96..... LD SP,(£FA96) ==>** load Stack Pointer with the value held in memory location £FD96.

The stack is organised as a **LAST IN FIRST OUT [LIFO]** sequential structure, which means that the first piece of data placed on the stack is the **bottom of the stack**, and the current data value is at the **top of the stack**. Remember that the stack builds down in memory.... therefore the **top of the stack = smallest memory address & bottom of the stack = higher memory location**.

The CPU uses the stack to store **return addresses** when branching to a subroutine. The **Program Counter [PC]** keeps track of the instruction to be executed, and is automatically updated before the instruction is carried out.....The CPU updates the PC by adding the number of bytes in the current instruction to the value in the counter and this the points to the next instruction.

Memory address	code	meaning	:	Stack	High	Memory
5AF5	3620	LD (HL),£20	:	SP before call points here ==>	£00	£C000
5AF7	CD0D5B	CALL £5B0D	:	SP when Call is executed points to this location =====>	£FA	£BFFF
			:		£5A	£BFFE
5AFA	Next Instruction		:	Lower Memory		£BFFD

In the latter example, the PC points to address £5AF7. The instruction at that memory location is 3 bytes long, add these three bytes to £FA57 and the result is 5AFA. The CPU automatically puts this into the PC.

The CPU then detects that the instruction is a CALL and places the value on the STACK before the CALL is made. When the CPU encounters a RET instruction it takes the current value from the **top of the stack** and places into the Program Counter [Remember on the Z.80 we always work with LSB MSB format] and the computer then executes the next instruction following the CALL which in our example is at £5AFA. You should be able to see how disastrous it would be if the **return address** was destroyed or overwritten. When the return address has been popped off the stack the PC points to its original address before the call.

The programmer accesses the stack with Push & Pop instructions. Used carefully, these two instructions represent a really powerful tool. PUSH will place a value on the stack, and POP will retrieve a value from the stack. These two instructions can only be used with register pairs: AF, BC, DE, HL, SP, IX, IY.....

PUSH HL will save the contents of the HL register pair on the stack.

POP HL will place the value at the top of the stack into the HL registers

The two instructions are useful for getting a value from one register pair into another register set.

```

                                LD HL, £3C00
                                LD B,H      ; Put value of H int
                                LD C,L      ; Do same with reg C
Or .....LD HL,£3C00
                                PUSH HL     ; Put HL on Stack
                                POP BC      ; HL now in BC

```

The last three lines of code do exactly the same thing and in a more compact way. Of course, there is nothing wrong with writing the code the first wayas long as it works !

Pushes & Pops can generally only be used on the same level. For example:

```

                                PUSH HL     ; Save value of HL
                                CALL DELAY ; Branch to subroutine
                                POP HL      ; Get value back

```

This method is correct. The return address will be the last value on the stack so it follows it will be the first value popped off by the CPU to find the return address. HL will then pop its original value from the stack.

```

This is wrong!.....PUSH HL
                                CALL DELAY
                                rest of program
                                DELAY: POP HL
                                OTHER CODE
                                RET

```

The above code would lead to a program crash. The HL registers would have popped the RET address off the stack, and the program would eventually try to return to whatever value was placed on the stack with the PUSH HL instruction. You can, however, use this method to good advantage in some of your programming. But you must be aware of what values are held on the stack. Pushes leave the value of the registers unchanged.

PAGE2 An Introduction to BIORHYTHMS

This section of the program sets out to explain, what BIORHYTHMS are, their implications and how to interpret them

From the day of birth EVERYONE goes through cyclic patterns.

There are 3 known cycles :-

1. PHYSICAL every 23 days

2.EMOTIONAL . every 28 days

1 3.INTELLECTUAL every 33 days

The following example shows a graph of an EMOTIONAL cycle,

PROG2

```
#DISPLAY PAGE10.  
#ENTER  
#DISPLAY PAGE11.  
#ENTER  
#DISPLAY PAGE12.  
#ENTER  
#DISPLAY PAGE13.  
#ENTER  
#RETURN
```

PAGE3

```

HIGH | eee < -----HIGH-----> eee
+100 | e e < -----PEEK-----> e e
      | e e e
      |
      | e e
      |
      | e e
      |
      |
      | DAYS ----->
0 | -1--5---x0---15---20---x25---30-
    ^
    ^
    | Today e e
    |
    | e e
    |
    | e e
    |
    | e e
    |
    | CRITICAL ee ee < LOW
-100 | DAYS e < PEEK
LOW MARKED x

```

PAGE1 BIORHYTHMS

(c) Woodensoft 1984

There follows a brief discription
and explantion of BIORHYTHMS.

Should you wish to miss this
section of the program then press
KEY 'C'

WHEN YOU HAVE FINISHED READING THE

DISPLAYED PAGE PRESS

'RET'

PAGE4

On the previous graph several key areas were marked

1. Today

Today's Biorythm value is
is shown nearest the left hand side

2.Critical days

Critical days occur as the cycle changes sign (crosses the zero line), these are shown as 'x' when using the 'all curves' option if two curves are critical on the same day that point is shown as 'x'.

On birth all cycles are critical, this does not occur again for 58 years, but when it does it is shown as '0'.

PAGES

3.Vertical scale

Biorhythms, when calculated return a value of between +100 through 0 to -100 they have no units.

4. High peak

This is where the cycle is at it's uppermost point and where you will benefit most. While on a high section you will feel:

Physical : Fit, Strong and Alert

Emotional : Able to cope easily
with an emotional
crisis.

Intellectual : Mentally Alert,able
to solve problems
and creative.

PAGE6

5.Low troughs

During this period
you are at a low ebb,you feel:

Physical : Tired,Lazy & have slow
reactions.

Emotional : Depressed

Intellectual : Lack of ideas,and
prone to ridiculous
mistakes

PAGE7 6.

Critical:This is when you have to
be most carefull,on these days
(approximately 1 in 5) anything
can happen.You may feel:

Physical : Tired,your reactions
and stamina become
unpredictable,you suffer
fatigue & make bad
decisions.

Emotional : Touchy & irritable,all
decisions based on
emotion.

Intellectual : Unpredictable
judgement.

TAKE CARE

To read intro again press 'A'

PAGE10

Introduction to commands.

By entering the numbers between

1 & 8 the computer will perform

several different functions.

These functions are explained

in the following pages.

Press 'RET'

PAGE11

Explanation of commands

1. This section plots and displays
the 3 cycles of life.

The Physical cycle is drawn using
the letter 'p',and is based on a
23 day cycle.

The Emotional cycle is drawn
using the letter 'e',and is based
on a 28 day cycle.

The Intellectual cycle is drawn
using the letter 'i',and is based
on a 33 day cycle.

PRESS 'RET'

PAGE12

2. This section will change the
date entered in the machine
to the required test date,
Dates must be entered as

DD/MM/YY

3. This is as above but effects
the date of birth of the
subject.

PRESS 'RET'

PAGE13

4. Produce the EMOTIONAL
cycle graph.

5. Produce the INTELLECTUAL
cycle graph.

6. Produce the PHYSICAL
cycle graph.

7. Display todays values for
the EMOTIONAL,PHYSICAL &
INTELLECTUAL cycles.

8. Command explanation
(this section)

9. List dates on which CRITICAL
days will occur in the next
30 days.

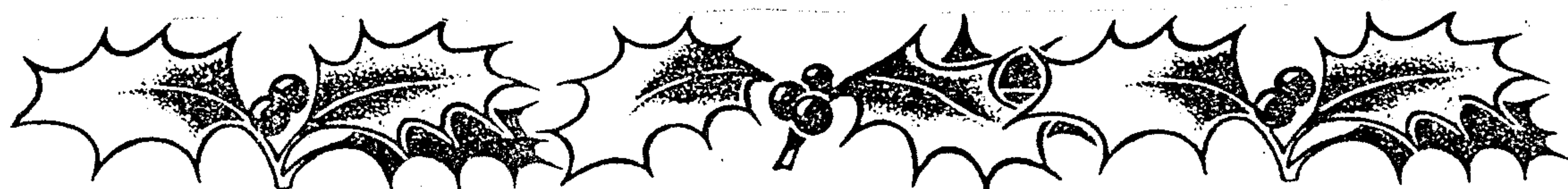
PRESS 'RET'

**** THE MAN FROM
***** G.R.A.N.N.Y

A massive adventure
that is too big (sorry)
for the MTX 500.....
A fantastic adventure..

James Bond eat
your heart out...

£5.95 from the
Club's own
table..Syntax.



MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

```

10 GOSUB 2130
20 PLOT "PROG1"
30 REM *****
40 REM ** MENU **
50 REM *****
60 LET Q=0: LET S=Q: LET U=S
70 PAPER 11
80 INK 1
90 CLS
100 CSR 14,0: PRINT "BIORHYTHMS"
110 CSR 14,1: PRINT "*****"
120 CSR 2,4: PRINT "Today's date :";C$
130 CSR 2,6: PRINT "Date of Birth:";D$
140 CSR 16,8: PRINT "MENU."
150 CSR 5,10: PRINT "1. Plot all graphs."
160 CSR 5,11: PRINT "2. Change today's date."
170 CSR 5,12: PRINT "3. Change date of birth."
180 CSR 5,13: PRINT "4. Display EMOTIONAL cycle."
190 CSR 5,14: PRINT "5. Display INTELLECTUAL cycle."
200 CSR 5,15: PRINT "6. Display PHYSICAL cycle."
210 CSR 5,16: PRINT "7. Display today's values."
220 CSR 5,17: PRINT "8. Explain Commands."
230 CSR 5,18: PRINT "9. Find CRITICAL days."
240 CSR 8,22: INPUT "Enter code to continue.";T
250 IF T<0 OR T>9 THEN GOTO 30
260 ON T GOTO 30,280,450,780,900,1110,1280,1450,1580,1600
270 GOTO 270
280 REM *****
290 REM ** ALL GRAPHS **
300 REM *****
310 GOSUB 2030
320 LET Z=1
330 GOSUB 940
340 GOSUB 1150
350 GOSUB 1320
360 LET Z=0
370 GOTO 380
380 REM *****
390 REM ** PAUSE **
400 REM *****
410 CSR 7,22
420 PRINT "Press any key to continue"
430 IF INKEY$="" THEN GOTO 430
440 GOTO 30
450 REM *****
460 REM ** CHANGE TODAY'S DATE **
470 REM *****
480 CLS
490 CLS : CSR 1,10: PRINT "Enter today's date (DD/MM/YY) "
500 GOSUB 570
510 LET C$=A$
520 LET R=D
530 LET C$(3)="/"
540 LET C$(6)="/"
550 LET G=R-F
560 PAUSE 2000: GOTO 30

```

```

570 REM *****
580 REM ** INPUT DATE **
590 REM *****
600 INPUT A$
610 IF LEN(A$)<>8 THEN GOTO 600
620 FOR N=1 TO 8
630 IF N=3 OR N=6 THEN NEXT N
640 IF A$(N)<"0" OR A$(N)>"9" THEN GOTO 600
650 NEXT
660 LET A=VAL(A$(7,2))
670 LET B=VAL(A$(4,2))
680 LET C=VAL(A$(1,2))
690 IF B=0 OR B>12 THEN GOTO 570
700 IF C=0 OR C>(ASC(B$(B)))/2 THEN GOTO 570
710 IF B=2 AND C=29 AND NOT (A/4=INT(A/4)) THEN GOTO 570
720 LET D=(C-1)+(A*365)+(INT(A/4))
730 IF B=1 THEN RETURN
740 FOR N=2 TO B
750 LET D=D+(ASC(B$(N-1)))/2
760 NEXT
770 RETURN
780 REM *****
790 REM ** CHANGE DATE OF BIRTH **
800 REM *****
810 CLS
820 CSR 1,10: PRINT "Enter date of birth (DD/MM/YY) "
830 GOSUB 570
840 LET D$=A$
850 LET F=D
860 LET D$(3)="/"
870 LET D$(6)="/"
880 LET G=R-F
890 PAUSE 2000: GOTO 30
900 REM *****
910 REM ** PLOT EMOTIONAL GRAPH **
920 REM *****
930 GOSUB 2030
940 LET X=6
950 FOR N=6 TO G+30
960 LET E=INT(101*SIN(2*PI*(N/28-INT(N/28))))
970 IF ABS(SGN(U)-SGN(E))=2 THEN LET E=0
980 LET U=E
990 LET Y=INT(10-E/10)
1000 CSR X,Y+1
1010 IF Y=10 THEN GOSUB 1070: GOTO 1030
1020 PRINT "e"
1030 LET X=X+1
1040 NEXT N
1050 IF Z=1 THEN RETURN
1060 GOTO 380
1070 IF SPK$="x" THEN CSR X,Y+1: PRINT "x": RETURN
1080 IF SPK$="z" THEN CSR X,Y+1: PRINT "0": RETURN
1090 CSR X,Y+1: PRINT "x"
1100 RETURN
1110 REM *****
1120 REM ** PLOT INTELLECTUAL GRAPH **
1130 REM *****

```

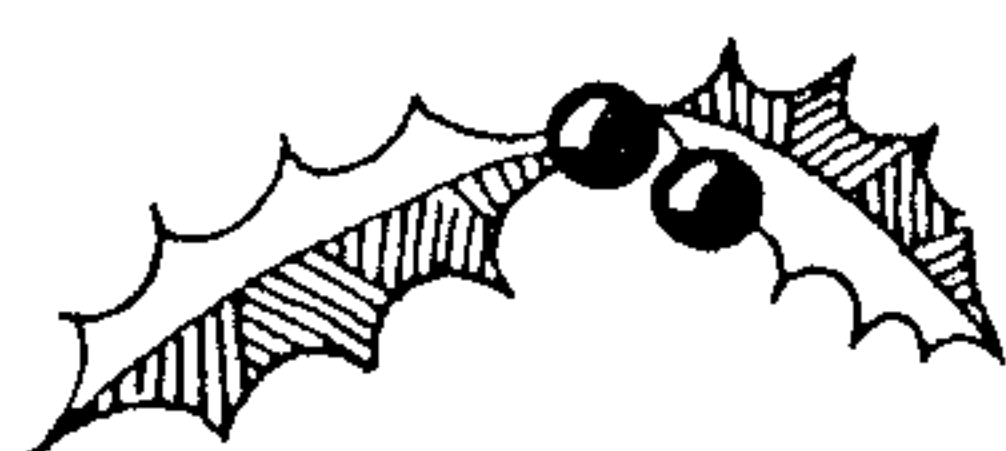


```

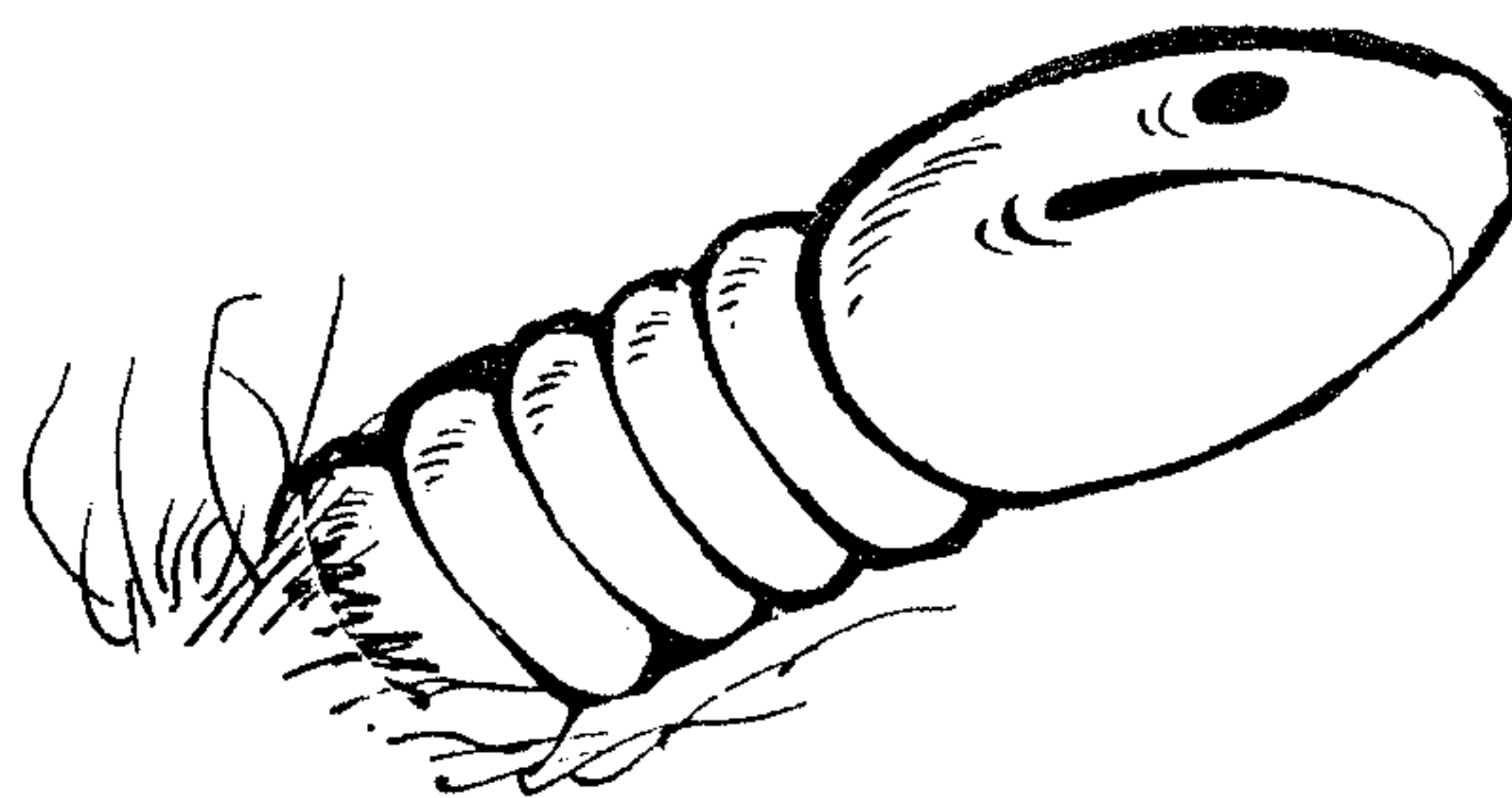
1710 IF NOT (P=0 OR I=0 OR E=0) THEN GOTO 1770
1720 CSR 2,W: PRINT E$
1730 IF P=0 THEN CSR 14,W: PRINT "P"
1740 IF E=0 THEN CSR 16,W: PRINT "E"
1750 IF I=0 THEN CSR 18,W: PRINT "I"
1760 LET W=W+1
1770 NEXT N
1780 GOTO 380
1790 CLS
1800 CSR 12,1
1810 LET E=(100*SIN(2*PI*(N/28-INT(N/28))))
1820 IF ABS(SGN(U)-SGN(E))=2 THEN LET E=0
1830 LET U=E
1840 LET I=(100*SIN(2*PI*(N/33-INT(N/33))))
1850 IF ABS(SGN(S)-SGN(I))=2 THEN LET I=0
1860 LET S=I
1870 LET P=(100*SIN(2*PI*(N/23-INT(N/23))))
1880 IF ABS(SGN(Q)-SGN(P))=2 THEN LET P=0
1890 LET Q=P
1900 LET A=VAL(C$(7,2))
1910 LET B=VAL(C$(4,2))
1920 LET C=VAL(C$(1,2))
1930 LET C=C+N-6
1940 IF C<=((ASC(B$(B)))/2)-((B=2) AND (A/4=INT(A/4))) THEN GOTO 2010
1950 LET C=C-((ASC(B$(B)))/2)-((B=2) AND (A/4=INT(A/4)))
1960 LET B=B+1
1970 IF B<=12 THEN GOTO 1940

```

MEMOPAD 20



program listing



This listing was sent in by Mr. B. W. Brown. In the program you guide Willie Worm around the garden eating mushrooms, flowers and bugs. Be careful you don't run into the walls or turn onto yourself as you get bigger and bigger.

```

1 LET Z=0
2 LET SC=0
3 VS 4: CLS
4 DIM SC(12),SC$(12,50)
5 FOR T=1 TO 10: LET SC(T)=0: LET SC$(T)=".....": NEXT
6 LET H=0
7 GOSUB 20000: GOSUB 35000
8 LET C=0: CLS
9 LET H=0
10 DIM A1(90,90): DIM B1(40,30)
20 LET S=0: LET J=0
21 VS 4: CLS
22 LET Z=0
30 LET X=17: LET Y=12
31 LET A=X: LET B=Y
32 LET X1=0: LET Y1=0
33 CLS
36 LET J=J+10
37 LET F=S
38 COLOUR 4,1
40 COLOUR 3,11
41 COLOUR 2,10
42 COLOUR 0,10
43 COLOUR 3,1
44 VS 4: CLS
60 FOR T=0 TO 7
70 LINE T*8,18,T*8,183: LINE 240+T,18,240+T,183
71 LINE 8,16+T,247,16+T: LINE 10,176+T,244,176+T
80 NEXT T
131 COLOUR 3,11
139 COLOUR 1,6
140 FOR I=1 TO J
141 RAND -13
150 LET Q=INT(RND*28+2): LET W=INT(RND*20+2)
160 CSR Q,W: IF ASC(SPK$)<>32 THEN GOTO 150
170 CSR Q,W: PRINT "b"
180 NEXT I
190 LET E=0
200 LET DI=3
210 LET Q=S
211 COLOUR 1,1
220 CSR 2,0: PRINT "SCORE:": CSR 8,0: PRINT S

```

```

230 CSR 15,0: PRINT "HI-SCORE:": CSR 24,0: PRINT H
240 CSR 1,22: PRINT "STRENGTH:": CSR 10,22: PRINT E
300 LET A1(X,Y)=X1
310 LET B1(X,Y)=Y1
320 LET X=X+X1
330 LET Y=Y+Y1
331 IF X<2 OR X>29 OR Y<2 OR Y>20 THEN GOTO 10000
332 CSR X,Y: LET AS=ASC(SPK$)
340 IF AS=32 THEN GOTO 400
350 IF AS=97 THEN GOTO 10000
360 IF AS=101 AND E=0 THEN GOTO 10000
365 LET Q=S
370 IF AS=98 THEN GOTO 2000
380 IF AS=101 THEN GOTO 2200
390 IF AS=99 THEN GOTO 2300
395 IF S-F=J*70 THEN GOTO 30
400 CSR X,Y: PRINT "a"
401 LET Q=Q+10
420 IF Q<5 THEN GOTO 500
430 LET C=A1(A,B)
440 LET B=B1(A,B)
450 LET A=A+C
460 CSR A,B: PRINT " "
500 LET X$=INKEY$: SOUND 1,INT(RND*30+100),5
505 IF X$="" THEN GOTO 300
510 IF X$="Z" OR X$="X" THEN GOTO 511 ELSE GOTO 300
511 LET DI=DI+(X$="Z")-(X$="X")
520 IF DI>3 THEN LET DI=0
530 IF DI<0 THEN LET DI=3
540 LET X1=0: LET Y1=0
550 LET X1=X1+(DI=2)-(DI=0)
560 LET Y1=Y1+(DI=3)-(DI=1)
580 GOTO 300
2000 LET S=S+10
2010 CSR 8,0: PRINT S
2011 FOR T=400 TO 1 STEP -20: COLOUR 1,15: CSR X,Y:
2015 PRINT "f": COLOUR 1,1: SOUND 0,T,12: NEXT T: SOUND 0,0,0
2020 FOR G=1 TO 2
2021 RAND -19
2030 LET L=INT(RND*28+2)
2040 LET LL=INT(RND*20+2)

```




```

2050 CSR L,LL: IF ASC(SPK$)<>32 THEN GOTO 2030
2060 IF 6=1 THEN CSR L,LL: COLOUR 1,4: PRINT "c": NEXT 6
2065 CSR L,LL: COLOUR 1,12: PRINT "e": COLOUR 1,1
2070 NEXT 6
2071 CSR X,Y: PRINT " "
2100 GOTO 395
2200 LET S=S+40
2201 COLOUR 1,15
2202 FOR T=2 TO 400 STEP 20
2205 CSR X,Y: PRINT "g"
2206 SOUND 0,T,12: SOUND 1,T+2,12: NEXT T
2209 SOUND 0,0,0: SOUND 1,0,0: COLOUR 1,1
2210 CSR 8,0: PRINT S
2220 LET E=E-1
2221 COLOUR 1,1
2225 CSR 10,22: PRINT E;" "
2240 GOTO 395
2300 LET S=S+20
2301 FOR T=2 TO 400 STEP 20: COLOUR 1,15: CSR X,Y: PRINT "h"
2302 SOUND 0,INT(RND*300+200),12: NEXT
2303 SOUND 0,0,0
2305 COLOUR 1,1
2310 CSR 8,0: PRINT S
2320 LET E=E+1
2330 CSR 10,22: PRINT E;" "
2340 GOTO 395
10000 FOR T=0 TO 12 STEP .05: SOUND 3,2,15-T: SOUND 1,700,15-T
10025 NEXT T
10050 SOUND 3,0,0: SOUND 1,0,0: LET ZZ=100
10100 LET C=B1(A,B)
10110 LET A=A+A1(A,B)
10120 LET B=B+C
10125 SOUND 0,ZZ,12: SOUND 1,ZZ+20,12: LET ZZ=ZZ+10
10130 CSR A,B: IF ASC(SPK$)<>97 THEN GOTO 10240
10140 CSR A,B: PRINT " "
10150 GOTO 10100
10240 SOUND 0,0,0: SOUND 1,0,0
10245 PAUSE 3000
10265 IF S>H THEN LET H=S
10290 VS 4: CLS : VS 5: PAPER 1: PAUSE 900: VS 4: CLS : 60SUB 40000: GOTO 20
20000 GENPAT 0,97,28,63,121,121,121,121,30,0
20010 GENPAT 0,98,28,62,127,127,8,8,8,0
20020 GENPAT 0,102,0,78,81,81,81,78,0,0
20030 GENPAT 0,103,134,137,169,233,169,38,0,0
20040 GENPAT 0,104,102,153,25,41,73,246,0,0
20050 GENPAT 0,101,60,36,153,126,126,126,126,129
20060 GENPAT 0,99,36,90,189,189,90,44,8,7

```

```

30000 RETURN
35000 COLOUR 0,1: COLOUR 1,3: COLOUR 2,4: CLS : CSR 4,4
35001 PRINT "*** WILLIE WORM ***": COLOUR 0,4
35002 COLOUR 1,15
35010 CSR 1,6: PRINT " GUIDE WILLIE WORM AROUND THE"
35020 CSR 1,8: PRINT " SCREEN USING Z AND X KEYS"
35030 CSR 1,14: PRINT " EAT MUSHROOMS b FOR 10 POINTS."
35040 CSR 1,16: PRINT " EAT FLOWERS c FOR 20 POINTS."
35060 CSR 1,18: PRINT " EATING FLOWERS ENABLES YOU TO"
35070 CSR 1,20: PRINT " EAT BUGS e 40 POINTS"
35100 CSR 1,22: PRINT " PRESS ANY KEY";
35110 LET K$=INKEY$: IF K$="" THEN GOTO 35110
35115 PRINT CHR$(7): PLOD "PROG": RETURN
40000 LET Z=0: FOR X=1 TO 10: IF S>SC(X) THEN LET Z=X: LET X=11
40001 NEXT X
40010 IF Z=0 THEN CLS : GOTO 40061
40020 CLS : CSR 2,2: PRINT " ENTER NAME FOR SCORE TABLE"
40030 CSR 2,6: PRINT ";: INPUT A$
40040 IF Z=10 THEN GOTO 40060
40050 FOR X=9 TO Z-1 STEP -1: LET SC(X+1)=SC(X): LET SC$(X+1)=SC$(X): NEXT X
40060 LET SC$(Z)="" : LET SC(Z)=S: LET SC$(Z)=A$+" "
40061 COLOUR 2,1: COLOUR 0,1: CLS
41000 CSR 2,1: PRINT "WILLIE WORM HALL OF FAME"
41005 FOR T=1 TO 10
41010 CSR 2,1+(T*2): PRINT SC(T),;SC$(T)
41020 NEXT T
41030 CSR 5,22: PRINT "PRESS ANY KEY TO PLAY"
41035 LET P$=INKEY$
41040 IF P$="" THEN COLOUR 1,INT(RND*13+2): GOTO 41000
41050 GOTO 20

```



PETER GOODE'S
MTX PROGRAM BOOK

We now have new stocks of
this excellent book. Some
very unique programs for
you to type into your MTX.

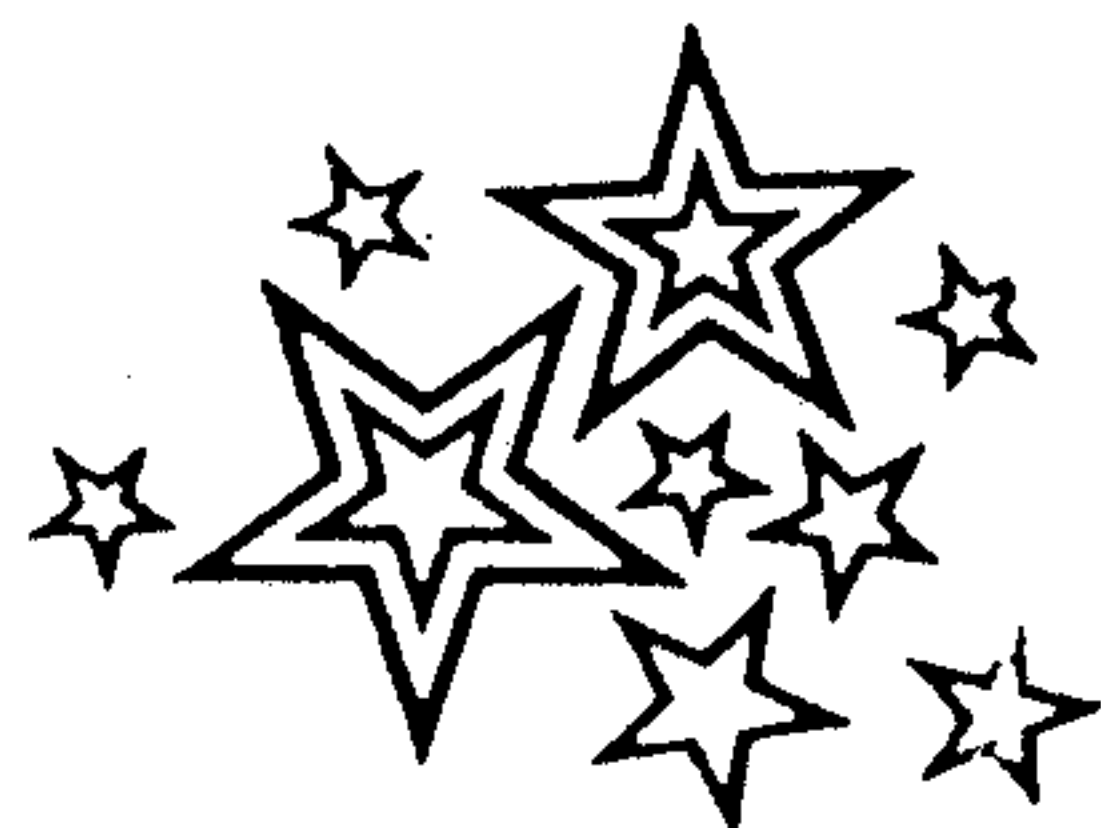
Price to Genpat
£5.25 inc p.p.



IMPORTANT NOTICE FOR ALL MEMBERS WHO ORDERED TRI-COM SOFTWARE.

TRI-COM'S new address is: 31,Warneford Road, Cowley, Oxford.

Any member who has ordered software from this company can telephone Tri-Com on Oxford 0865-248208. I spoke to Francis Wallinger 14 days ago and he assured me that he was doing everything within his power to resolve the duplicating problems.



 SOFTWARE LIST

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

 Price : a=£4.95 b=£5.95 c=£6.95 d=£7.95 e=£8.95 f=£9.95 g=£15
 h=£25 i=£4.50 j=£19.50

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

 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.....[AR, 02, c, I]
 DENNIS

 & THE CHICKEN...[AR, //, c, U]
 MISSILE KOMMAND....[AR, //, a, U]
 MAXIMA.....[AR, 01, c, I]
 M CODER.....[UT, //, c, U]
 JOHNNY REB.....[TW, //, c, E]
 THE KEY TO TIME....[AD, //, c, E]
 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, //, f, I]
 LORDS OF TIME.....[AD, //, f, I]
 COLOSSAL ADVENTURE.[AD, 02, f, I]
 SPELLI-COPTER.....[ED, //, b, I]
 UTILITIES 1.....[UT, //, a, I]
 COMPOSER.....[UT, //, g, 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, E]
 AGROVATOR.*.....[AR, 04, b, I]

 PURCHASE LEDGER.....[BS, //, h, E]
 BASIC BUSINESS.....[BS, //, b, E]
 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]
 GAUNLET.....[AR, //, c, I]
 COBRA.....[AR, //, c, E]
 MURDER AT THE MANOR....[AD, //, c, E]
 FRANTIC FREDDIE.....[AR, //, c, E]
 DRAUGHTS.....[BG, //, d, I]
 3D TACHYON FIGHTER....[AR, //, 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, //, b, I]
 TUMBLEDOWN TOWER.....[AD, //, i, E]
 EDASM.....[UT, //, d, I]
 DOODLEBUG DESTROYER....[AR, //, a, I]
 GRAPHICS.....[UT, 01, c, I]
 MAN FROM GRANNY.*.....[AD, //, a, E]
 ALICE IN WONDERLAND....[AD, //, c, I]
 HAWKWARS.....[AR, //, a, E]
 BOUNCING BILL.....[AR, //, a, E]

* runs only on the MTX512 and RS128

HACKERS TIP by STEPHEN WHITE

By using the following information you can obtain extra lives in Snappo. Just a word of warning - if you choose to make this alteration it does tend to take a lot of fun out of the game because there is no challenge left.

Follow the instructions below depending upon your machine.

LOAD "SNAPPO", then RESET the machine and type.

MTX500

POKE 40861,no. of lives (0-255)
POKE 64164,88
POKE 64165,164
POKE 64167,88
POKE 64168,164
POKE 64172,88
POKE 64173,164
VS 4: CLS: RUN

MTX512

POKE 24477,no. of lives (0-255)
POKE 64164,88
POKE 64165,64
POKE 64167,88
POKE 64168,164
POKE 64172,88
POKE 64173,64
VS 4: CLS: RUN

To save the program with the above alteration, type GOTO 500 instead of VS 4: CLS: RUN'.

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

Sorry the high scores were not printed in the last issue but there was no room.

Richard Franks sent in these high scores.

SNAPPO	107,450	Level 10
KNUCKLES	48,000	Level 8
TAPEWORM	40,260	Level 9
TOADO	25,486	Level 7

Maxima seems popular with people striving for high scores. Just look at these.

CONRAD HOUGHTON	69,500
B. CONWAY	101,600
DALJINDER SINGH	159,000....1hr 50 mins to manage that score

The following scores are still intact from Issue 2.

STAR COMMAND	52,250	DEREK WHITE.....Mission completed.
PHAID	23,470	ERIC PETERS.

Send those high scores for any game you have and we'll add it to our list.

program listing

FEELING LEFT OUT IN THE COLD ? THEN HAVE A GAME OF "SOLITARE" .
The instructions are contained within the listing and are self explanatory....



```

2 CRVS 2,0,1,19,32,40 REM" by J. & D.H. Software 1984, for MEMOPAD"
1 LET Q=0: LET S=48: POKE 64145,128
2 CRVS 2,0,1,19,32,4,0: PAPER 5: INK 15: VS 5: GOTO 810
3 LET F=0: LET T=0: LET R=0: LET L=1
4 VS 5: CLS
10 PRINT "          SOLITARE"
11 PRINT "          -----": PRINT
15 DIM A(64)
30 PRINT "48 Checkers are placed on the 2 outside"
40 PRINT "spaces of a standard 64 square checker-board. The object of the game is to  remove as many checkers as possible by"
50 PRINT "diagonal jumps. Use the grid reference to select the desired move."
51 PRINT : PRINT "As an example to move the top left hand corner piece type A1.": PRINT : PRINT "The only valid move for this is to C3"
52 PRINT : PRINT "If you cannot make any more moves type 'END'"
53 VS 2: CLS : INPUT "press 'RET' to continue ";S$
210 FOR J=1 TO 64
220 LET A(J)=1
230 NEXT J
240 FOR J=19 TO 43 STEP 8
250 FOR I=J TO J+3
260 LET A(I)=0
270 NEXT I
280 NEXT J
290 LET MM=0
295 VS 5: CLS
300 GOTO 510
310 VS 2: PAPER 5: INK 15: CLS : INPUT "JUMP FROM ";X$
311 GOSUB 1000: LET F=X
315 IF F<9 THEN LET R=1 ELSE LET R=INT(F/8)+1: IF MOD(F,8)=0 THEN LET R=R-1
316 IF F<9 THEN LET L=F ELSE LET L=MOD(F,8)
317 IF L=0 THEN LET L=8
318 IF R=9 THEN LET R=8
320 IF F=0 THEN GOTO 700
325 INPUT "TO ";X$: GOSUB 1000: LET T=X
329 IF T<9 THEN LET R1=1 ELSE LET R1=INT(T/8)+1
330 IF MOD(T,8)=0 THEN LET R1=R1-1
332 IF T<9 THEN LET L1=T ELSE LET L1=MOD(T,8)
333 IF L1=0 THEN LET L1=8
334 IF R1=9 THEN LET R1=8
340 PRINT
350 LET F1=INT((F-1)/8)
360 LET F2=F-8*F1
370 LET T1=INT((T-1)/8)
380 LET T2=T-8*T1
390 IF F1>7 THEN GOTO 460
400 IF T1>7 THEN GOTO 460
410 IF F2>8 THEN GOTO 460
420 IF T2>8 THEN GOTO 460
430 IF ABS(F1-T1)<>2 THEN GOTO 460
440 IF ABS(F2-T2)<>2 THEN GOTO 460
450 IF A((T+F)/2)=0 THEN GOTO 460
452 IF A(F)=0 THEN GOTO 460
454 IF A(T)=1 THEN GOTO 460
456 GOTO 500
460 VS 2: CLS : PAPER 5: INK 15
465 CLS : PRINT "ILLEGAL MOVE... Try again..."
466 PAUSE 3000: VS 2: CLS
470 GOTO 310
500 GOTO 900
510 PRINT "          ";;: FOR J=1 TO 8: PRINT J;" ";;: NEXT : PRINT : PRINT
511 LET Z=ASC("A"): FOR J=1 TO 57 STEP 8
512 PRINT "          ";CHR$(Z);" ";;: LET Z=Z+1
515 FOR I=J TO J+7

```

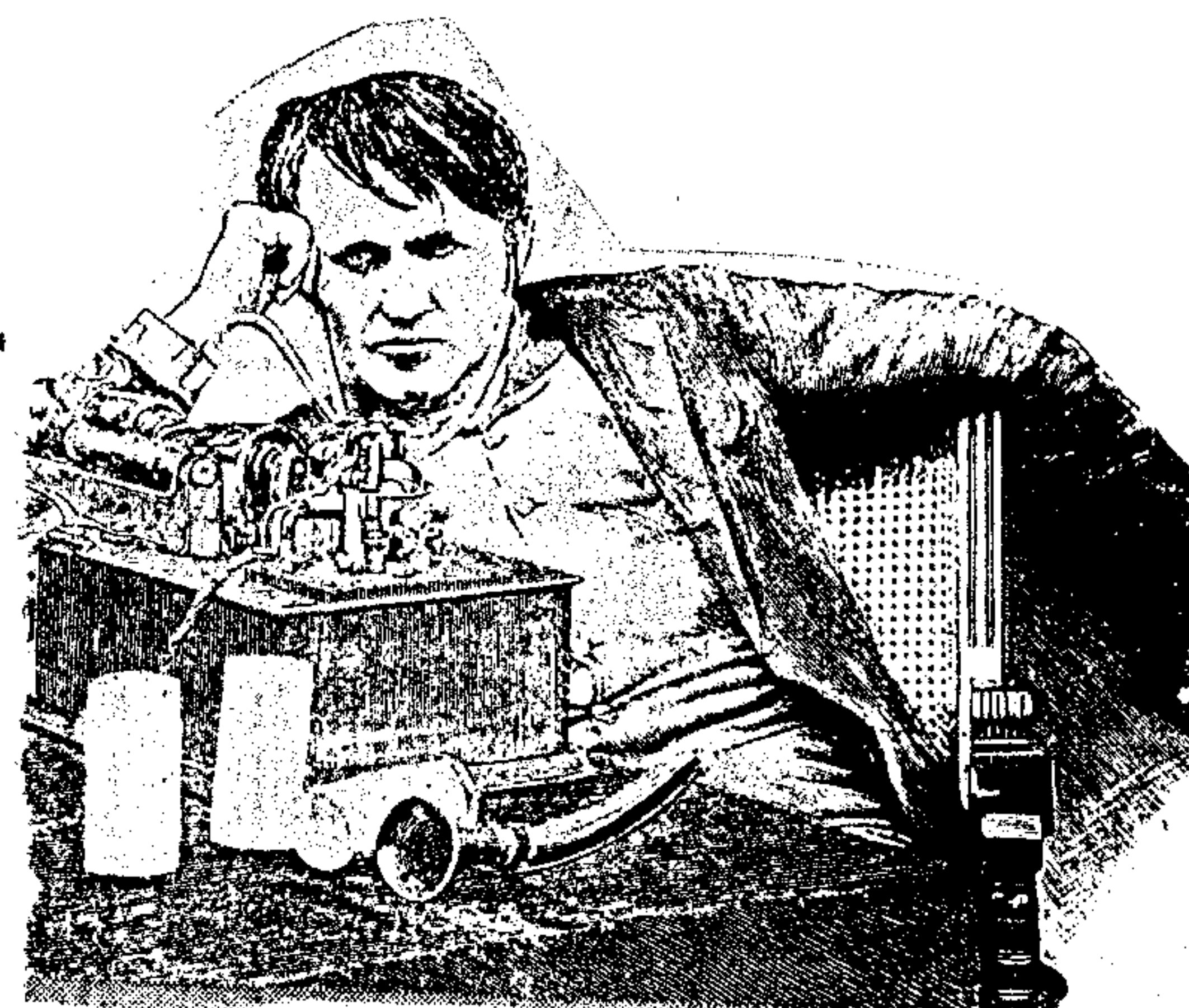


MEMOPAD the official magazine of GENPAT - Memotech MTX User Club.

```

520 IF A(I)=1 THEN PRINT CHR$(147); " "; ELSE PRINT ". ";
525 NEXT I
530 PRINT : PRINT
535 NEXT J
540 PRINT
545 GOTO 310
700 REM
740 VS 5: CLS : PRINT " You made ";MM;" jumps and had ";S;" pieces left on the board."
760 INPUT " Try again? ";A$
765 VS 5
770 IF LEFT$(A$,1)="y" OR LEFT$(A$,1)="Y" THEN GOTO 210
780 STOP
800 CLS
810 GENPAT 1,147,112,248,248,248,248,112,0
820 GOTO 3
900 VS 5: LET R=2+R*2: LET L=6+(L*3): CSR L,R-2: PRINT ". "

```



```

910 LET MM=MM+1: LET S=S-1
920 LET R1=2+R1*2: LET L1=6+(L1*3): CSR L1,R1-2: PRINT CHR$(147)
930 IF R>R1 THEN LET R=R-2 ELSE LET R=R+2
940 IF L>L1 THEN LET L=L-3 ELSE LET L=L+3
950 LET A(T)=1: LET A(F)=0: LET A((T+F)/2)=0
980 CSR L,R-2: PRINT ". "
990 VS 2: PAPER 5: INK 15: CLS
999 GOTO 310
1000 IF X$="END" THEN GOTO 700
1002 IF LEN (X$)>2 THEN GOTO 460
1005 LET K=ASC(LEFT$(X$,1))-64: LET M=VAL(RIGHT$(X$,1)): LET X=M+8*(K-1)
1010 IF K>8 OR K<=0 THEN GOTO 465
1020 IF M=0 OR M>8 THEN GOTO 465
1030 RETURN
2000 SAVE "SOLITARE": GOTO 0
2010 GOTO 0

```

Competition

Here's something to think about when you have a quiet moment during the festive season.

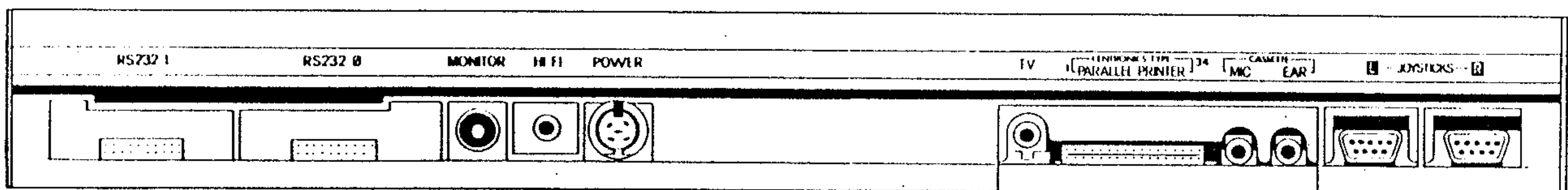
It is apparent from looking at programs submitted for publication in Memopad that most of you like to cram as much as possible onto one program line. This competition will be right up your street !

For the best ONE LINE program we will award two pieces of software and a MTX dust cover.

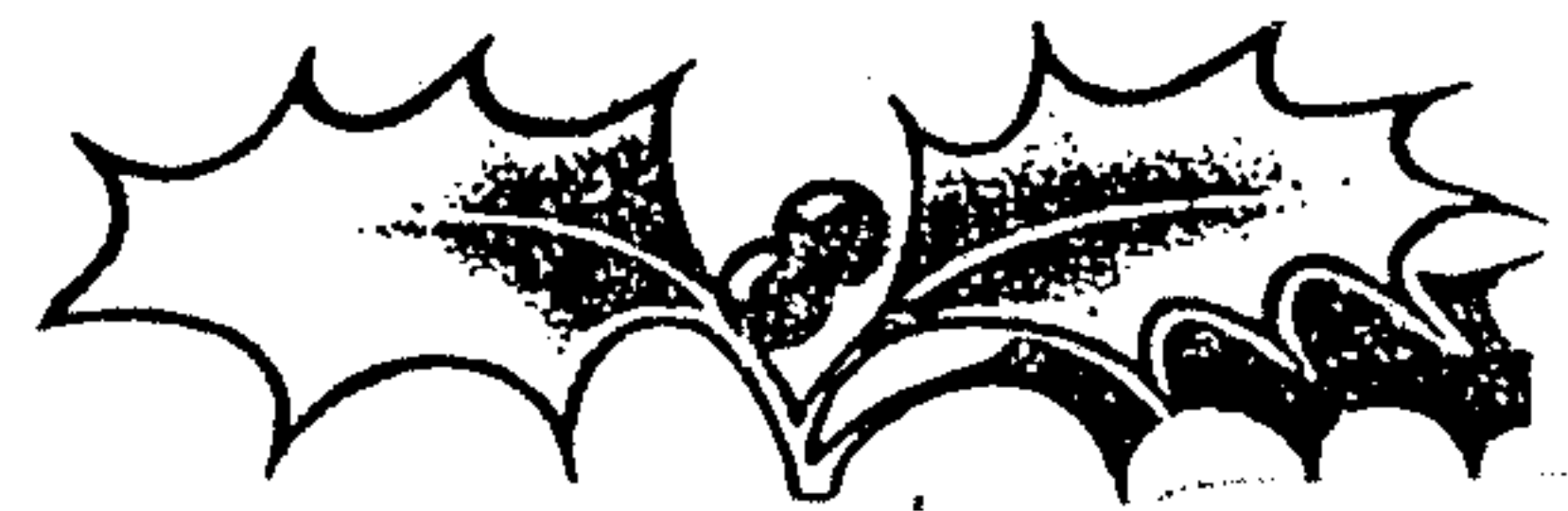
No restrictions are placed on the subject.... as long as the program is capable of doing something useful it will qualify for the competition. Your program need not be confined to one line, but we shall be judging the competition solely on what your one line actually does.

WINNER OF THE BEST PROGRAM SUBMITTED IN ISSUE TWO & THREE

The winner of the best Member Program competition run in issues two and three is Mr.D.A. Buck for his Screen Scroll Routine [issue two].



END STATEMENT



Christmas is here! It doesn't look as if we shall become the top selling computer of 1984, but never mind, there's always next year. It really is about time that Memotech realised what a good computer they have produced and started to yell it from the roof tops. If ever the time was ripe for the MTX to take over the British market, it is now ! No other computer manufacturer has produced anything sparkling this year, and nothing can compete with the engineering quality of the MTX. Amstrad managed to capture the bulk of the home market purely on the grounds of marketing, and that is something for the top echelon at Memotech to give serious thought to.

We have been absolutely swamped with orders over the past few weeks, and we have concentrated on dealing with these before anything else. Some people are bound to have been disappointed..... this is purely due to leaving their orders too late - stocks do run out !

On the subject of software, what has happened to Tri-com is beyond belief. The club managed to acquire titles, produce the masters, and deliver orders, all within three weeks! So Tri-com's "We are having difficulty getting duplications" just doesn't hold water. I have tried several times this week to obtain satisfaction from Francis Wallinger but as yet, I have not had any reply to my telephone calls. I do not want to create any bad feeling, but the message from the club to Tri-Com...GET YOUR ACT TOGETHER...FAST!

The CP/m article has been held over - yes! we're always holding things over - to get this magazine out I have had to work four days at the printers as an un-paid 'skivvy'. The article is finished and ready for publication. However, to make sure this edition is in the post before Xmas we have had to cut things a little short.

MTX owners are really laid-back: we have had 16 replies to the competition, the raffle is worse than last time, so it really is not worth our time and effort in producing these features in the magazine. After Christmas we shall dispense with them completely for the time being. One new feature will be the choice of purchasing the programs published in the magazine on tape for £3.00 including postage.

Any member wishing to contact other members can write into Genpat requesting that their name appears in the magazine. In this way you could possibly arrange your own local branches.

It only remains for me to thank you for your support in 1984, and to wish you, once again, A MERRY CHRISTMAS, and we will be back with you in 1985. Thanks also to all those members who have sent in their programs and tapes. Please keep sending your submissions... we depend on you.

HAPPY NEW YEAR from all at Genpat.

Chief Advisor to Genpat: Jeff Wakeford*Pascal Consultant:Stephen Varley*
Technical Consultant:Geoff Boyd [Memotech]*Software Consultant:Jim Wills
[Memotech]*Cover Design Mike Williams* Assistant to Editor:Tim Rothwell*
Sales & Software* Patricia. <c> GENPAT 1984