JUNF '88

Hi, its me again. How is it going with all you animals out there? Good I hope!
I have had a very positive responce as far as subscriptions are concerned. Thank
you to those special people out there!. I am still negotiating with various
people for special "things" for those special people. I think that I will keep
you in suspence for a while before I let you know what special offers you will be
able to enjoy in the comming months!

Mark Berkowits's MACHINE CODE articles are drawing to an end. I should very much
like to know what your verdict is on the series. Interesting, confuzing, boring,
not sure? well let me know! I would also very much like you to motivate your
answer please.

There are heaps of special offers in this newsletter, take advantage of them before
it is to late! Look out for the Special Offer Days on the Gemini, LC10 and Z88!
These items are only on special on those days!
There is also a continuation of the 1 off items that I.E. has left in stock -
remember its first come first serve - there is only one of each title in stock!
I have had the opportunity to look into PAPER PRICES the last month, to
discover that I.E. is the cheapest supplier around. I would strongly suggest that
you took advantage of their offers on paper!
PC and +3 owners will be pleased to see that there is something that should be of
interest to them. Tasmans programs are very good programs in my opinion and if
you are after printing utilities - then their series is very much a must!

For those enthusiasts in the Club who feel the cold of winter - like me! -
scarves and knitted caps are available with specialy designed logo knitted into
them! Come on and be different - there are only 20 scarves - so get yours NOW!
All scarves are knitted from the very best lambs wool and come in an asortment
of colours:- Grey, Red, Blue. Scarves only cost R30.00. support your club!

I have had a number of requests for articles to do with PC's. If you are a PC
owner and had something to do with the Speccie, let us know about your doings
and experiments with the two! Have you converted any PC software to run on the
Spectrum? Like to take it up as a challenge? Think about it and then let me
know! I have also had requests for articles to do with graphics and printing
on the Spectrum. Please can I beg SOME ALIEN in this galaxy to do me a small
favour, and just sit down for ½ hour and put an article together for me?!! Thanx
Remember - the club is for you the readers, and without your contributions then
the club would not exist! I really would not like that to happen.

I have gabbled enough for now. On with the newsletter!

Anne

MAN LOGIC

By NEIL BRADLEY

THOUGHTS & CROSSES

<u>BUY NOW BEFORE THE PRICE GOES UP!!!</u>

Unfortunately a good thing never seems to last very long, especially where prices are concerned!

The following hardware is set for a price increase in the near future. I.E. has presently got stocks of the hardware either on order from England or at their warehouse in Bophuthatswana.

The Pound / Rand exchange rate is now standing at R4.18 to £1, in other words any item costing £10 will cost you R41.80. This time last year the same item cost only R33.70! When you consider freight costs as well, then it costs more due to the Rand value, and the basic cost of freight went up on 1 May 1988 from £3.50 per Kilogram to £4.25 per Kilogram.

Things are certainly costing more and more. Take advantage of the temporary price freeze and buy your hardware <u>NOW</u>. Remember that as I said - a good thing never lasts forever!

We also forsee that new customs duties will be levied in line with the governments policy of reducing public spending. Don't let them get the better of you!

Item	Price	P&P
Spectrum + 2	599.00	10.00
Spectrum + 3	799.00	10.00
Spectrum 48K	279.00	10.00
Ram Print	159.00	5.00
ZXLPrint	149.00	5.00
Multiprint	159.00	5.00
Multiface 128	169.00	5.00
Multiface + 3	185.00	5.00

LIMITED SPECIAL OFFER

LAAGSTE PRYSE! BESTE AFSLAG!
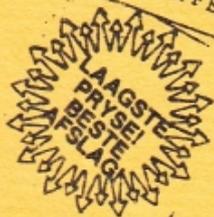
AMABOP INDUSTRIES and INFORMATION ENTERPRISES fall under the <u>CUSTOMS UNION OF SOUTHERN AFRICA</u>. They pay the same customs duties as similar import organizations throughout SA.

Their cheap prices relate to:-

* Good management,
* Good buying skills, and
* low mark-up.
                                        Thank you

Sir Clive has secured US distribution of the Z88 portable, in a deal worth $7.1 million for an initial 13,000 units!
The Z88 will be sold through Portland-based Diversified Foods, which is setting up a new (inedible?) company called Sinclair Systems Inc. The Z88 will be launched in May for $549.00.

COMPILATION TITLE	PRICE
KONAMI ARCADE COLLECTION	R42.45

Mike, Jail Break, Green Beret, Ninja Master, Shao Ian's Road, Nemisis, Hyper Sports, Ping Pong, Jackal, Yie Ar Kung Fu

ALL STATS	38.80

Xecutor, Inside Outing, Bobby Bearing, Shadow Skimmer, Mindstone, Warlock, Psytraxx, Brian Bloodaxe, That's the Sprit, Starbike

DATA EAST'S ARCADE ALLY	29.99

Kung Fu, Express Rider, Breakthru, Last Mission

ARCADE FORCE FOUR	42.45

Gauntlet, Road Runner, Metro Cross, Indiana Jones and the Temple of Doom

GAUNTLET LIMITED EDITION	42.45

Gauntlet, Gauntlet II

FAIRLIGHT LIMITED SPECIAL	26.99

Fairlight, Fairlight II, Fairlight the Legend

Z88 SPECIAL

For those of you who were not able to purchase the Z88 when the price was at its special Introductory offer, I have reached a special agreement with I.E. and am now able to offer the machine at the old price on three days this month!

Take advantage of this offer, as this offer is not to be repeated as a direct result of the poor Rand-Pound price.

20 JUNE
4 JULY
20 JULY

The Special Offer Days Are:-

Normal Price of Z88 - R999.00
Special Price   - 899.00

For more details about the computer, send a SAE + R1 to I.E., or simply look for details in the past few months mewsletters.

KNOCKOUT SAVINGS!

Are you a bit short of cash, but would like to impress your family and friends with your gigantic collection of software? Now is the chance to do it, because the following compilations are either in stock, or have been ordered from England and are expected to come into stock shortly. Just look at these great titles! Order yours now while I.E. HAS STOCKS. Avoid disappointment and make sure you get yours now!

COMPILATION TITLE	PRICE
ALL 30 GAMES	R25.50

Antics, Plantfall, Gridrunner, Bismark, Metor Storm, Jet Set Gortie, 3D Tunnel, Stockmarket, Fridge Frenzy, The valley, Demon Knight, Ned's garden, Space intruders, Fall of Rome, Draughts, Zasom, Detective 20 Tons, Chess player, Tubecube, Invasion, Escape, Lazerzone, Strontium Dog, Pyramania Micromouse, Dragons Bane, Canpet Capers, Blood 'n Guts

SOLID GOLD	33.79

Gauntlet, Ace of Aces, Leaderboard, Winter Games, Infiltrator

COMPILATION TITLE	PRICE
BEST OF ELITE vol 2	33.99

Paperboy, Ghosts 'n Goblins, space Harrier, BombjackII

HITPAC 6 vol 2	31.99

Into the Eagles Nest, Batty, Ace, Shockway-rider, Lightforce

FIVE STAR	27.30

Zoids, Equinox, 3 weeks in paradise, Back to school, spindizzy

MEGA HITS	25.99
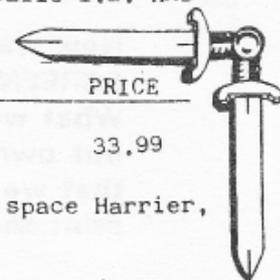
MACHINE CODE
PART 5

Now we start with the really interesting stuff; the setting up of the SCREEN information from the data we've just read in from tape. What we're going to do is to put all the neccessary characters into our own little 'dump' area (sort of like putting words into the spaces that we are going to provide), and then when we are done, we'll just print the whole lot in one small routine.

The SCREEN PICTURE AREA itself is going to lie in the addresses between where the tape header routine put the cassette information and the program i.e

ADDRESS	contents
29000-29017	Header information
29018-29091	Screen picture
29092-29288	The program

We'll worry about the complete contents of the Screen Picture Area later, what we're now concerned with is the moving of the Header Information to the spaces I have provided.

ENTRY CONDITIONS: HL points to the Tape Header Info
: DE points to the space in the Screen Picture Area
where I want certain words (or numbers,later)
to go.

EXIT CONDITIONS: A copy of the characters will be moved to the
: Screen Picture Area.

REGISTERS USED: A, HL, DE

A routine to move data around can be quite useful, and also be used quite often, so I am going to make it into a SUBROUTINE, and then have this next routine of mine CALL to it when I have set up the parameter information (see Entry Conditions).
The subroutine itself will come a little later in the program.

HEX CODE	ASSEMBLER MNEMONIC	COMMENT
3A4871	LD A,(29000)	: check first byte of Header
		: information.'0' = BASIC
A7	AND A	: Is it a '0'?
201B	JR NZ,T1	: if not, then jump ahead
116671	LD DE,29030	: otherwise, point DE at the
		: Screen Picture address...
CD2B72	CALL MV	: ..and CALL the move sub-
		: -routine to move...
50524F4752414D	DEFM "PROGRAM	: .. this word
00	DEFB 0	: (end of word indicator)
118771	LD DE,29063	: Do it again for...
CD2B72	CALL MV	
4C696E65	DEFM "Line	: ..this word...

4

```
00                   DEFB 0
1819                 JR   T2              :..and jump over the next bit
```

What the last section of program did was to move the words "PROGRAM"
and "Line" to the screen picture, because the header is for a BASIC
program.

This next section is for all other types of header, and is jumped
over by the program for BASIC programs.

```
HEX CODE             ASSEMBLER MNEMONIC        COMMENT

116671          T1   LD   DE,29030         : Address in the Screen Pic
                                           : area
CD2B72               CALL MV               : Move...
20204259544553       DEFM "  BYTES         : this word (NB 2 spaces!)
00                   DEFB 0
118771               LD   DE,29063         : ...and...
CD2B72               CALL MV
41646472             DEFM "Addr            : ..this one
00                   DEFB 0
```

When you see the sub-routine you'll see one way of moving bytes
around, but there are a few ways. Here is another:

```
HEX CODE             ASSEMBLER MNEMONIC        COMMENT

117471          T2   LD   DE,29044         : Space in the Screen Pic
                                           : area
214971               LD   HL,29001         : Characters in the Header
                                           : area
010A00               LD   BC,10            : Number of bytes to move
EDB0                 LDIR                  : .. and there you go
```

Quite a neat routine, is this one ! HL must contain the "FROM"
address, DE the "TO" address , and BC the number of bytes to move.
The information we're moving here, by the way, is the NAME of the
data block on cassette. However, if the name given on the cassette was
null (i.e ""), then this is going to contain the single byte #FF, and
nine blank spaces. (If you look in the Spectrum handbook, character
#FF is the word "COPY"). For the moment, though, this is not a problem
Let's have a look at numbers. Later you'll see the routine to
convert numbers to their ASCII values (I.E CODES 48 TO 57), so all
this routine does is set the pointer to where the number belongs,
loads HL with the two byte number, and call the sub-routine.
Once again, DE points to the spaces provided in the Screen Picture.

```
HEX CODE             ASSEMBLER MNEMONIC        COMMENT

2A5571          T3   LD   HL,(29013)       : This is the START address
                                           : or START line(if BASIC)
118D71               LD   DE,29069         : set the pointer
CD3572               CALL NM               : CALL the sub-routine

2A5371               LD   HL,(29011)       : This is the LENGTH
```

5

```
119D71                  LD   DE,29085        : set pointer again
CD3572                  CALL NM              : and CALL again
```

Alright, now for the final part of the main program: the printing
of the screen. As I mentioned earlier, the method I have used
allows you to use the "AT" function used in BASIC, and all this
does is simply print every character the is not a zero (0). The
first time it meets a zero byte, it will return to BASIC.

```
HEX CODE          ASSEMBLER MNEMONIC       COMMENT

215A71 .     LD    HL,29018         : The first byte of the
                                    : Screen Picture
7E       MORE LD   A,(HL)
A7            AND   A               : is it a zero ?
2804          JR    Z,EXIT          : if "yes" the exit..
D7            RST   16              : ...otherwise print it
23            INC   HL              : point to next byte
18F8          JR    MORE            : ...and do the same
C9    .  EXIT RET                   : RETURN TO BASIC
```

Now for the two sub-routines.

The first is the MOVE routine.

ENTRY CONDITIONS: HL points to "FROM" address
**   : DE points to "TO" address**

EXIT CONDITIONS: As soon as HL points to a "0" byte the routine
**   : is ended and control returned to where the CALL**
**   : originated.**

REGISTERS USED: HL, A, DE

```
HEX CODE          ASSEMBLER MNEMONIC       COMMENT

E1       MV   POP  HL               : This is the ADDRESS on the
                                    : stack to which the routine
                                    : will return, and it points
                                    : to the first byte of the
                                    : word we want to MOVE !
7E            LD   (HL),A           : So we just keep advancing
23            INC  HL               : it until we read a "0"..
E5            PUSH HL
A7            AND  A
C8            RET  Z                : ..and go back when it is
12            LD   (DE),A           : Put the byte at "DE"
13            INC  DE               : point at next address..
18F6          JR   MV               :.. and keep moving
```

The second routine is a little more complex. The Z80 cannot divide
two numbers directly, and one of the ways of doing this is to keep
subtracting until you split the number into groups of 10,000's ,
1,000's , 100's , 10's and units.
Believe me, it sounds harder than it really is !

6

	1	2	3	4	5	6	7	8
	R14.00	R18.00	R18.99	R19.50	R19.99	R24.99	R29.99	R39.99
A	Driller tanks	Driller tanks	Driller tanks	Driller tanks	Driller tanks	Driller tanks	Driller tanks	Driller tanks
	Horace goes sking	Horace goes sking	Horace goes sking	Horace hoes sking	Horace goes sking	Horace goes sking	Horace goes sking	Horace goes sking
	Horace and the spiders	Horace and the spiders	Horace and the spiders	Horace and the spiders	Horace and the spiders	Horace and the spiders	Horace and the spiders	Horace and the spiders
	Drive-in	Drive-in	Drive-in	Drive-in	Drive-in	Drive-in	Drive-in	Drive-in
	Backgammon	Backgammon	Backgammon	Backgammon	Backgammon	Backgammon	Backgammon	Backgammon
	Bubble Buster	Bubble Buster	Bubble Buster	Bubble Buster	Bubble Buster	Bubble Buster	Bubble Buster	Bubble Buster
	Antics	Antics	Antics	Antics	Antics	Antics	Antics	Antics
B	The Island, Space Raiders, Glider,	Casino Royal, Card Bonanza, Orc Attack,	Pyramid, Space Raiders, Drive-in, Bubble Buster, Rally Driver, Orc Attack, Island, Cargo, Magnets, Flippit	Run for Gold, Rally, 4 Quality Games,	Software Star, Rally Driver, Run for Gold, Orc Attack, Bubble buster, 4 Quality Games. Space Raiders	Magnets, Cargo, Glider, Flippit, Chess,	Rally Driver, Gift from the Gods, Island, AZTEC, hunt for the Sun Gold	Geometry, Snaffle, Chess Tutor, Word Wizard,
C	AZTEC, hunt for the Sun Gold	Video Pool, Backgammon	Dandy, Xcel, Thriller Pack	Twister, Football Manager,	W.O.W., Give my regards to Broad Street Camelot Worriors, Sabre Wulf	Collins Biology	Kayleth, Backgammon	Tasword 2, Lino-o-type

Look at these special offers! Here is a selection for everybody in the family from Junior the Mum.

All you have to do is choose one program from each block in the particular collum that you choose.

i.e. from collum 3, I would choose from block A - Driller Tanks
block B - Island
block C - Xcel

Three programs for the price of one!

All I would then have to pay for these three great programs is R18.99! Now is that value for money or what?

8

Spectrum lessons
At last MACHINE CODE!

A BEGINNER'S COURSE IN Z80 MACHINE CODE - part 3

By D. Roux

Here we go again. I believe last month's problem involved exchanging
the contents of two addresses. The syntax is reasonably straightforward:

```
LD A,(address1)
LD HL,address2
LD B,(HL)
LD (HL),A
LD HL,address1
LD (HL),B
RET
```

Remember: the brackets mean <u>content of</u>, so LD A,(address1) means load A
with the content of the first address, while LD HL,address1 means load
HL with the first address. It is OK to use a single register, like A,
to store the content of an address, since this can never exceed 255, but
I had to use a register pair (HL) to store the address itself, since
this can be a lot larger than 255!

Also, I could say LD A,(address1), but not LD B,(address2): I had to use
a rather roundabout method there, saying LD HL,address2 and then
LD B,(HL). The reason for this is because A, as I said last month, is
a <u>favoured register</u>. I think I should give a list of LD instructions
possible with A and with other registers:

LDs possible with A	LDs possible with any other register
LD A,(BC)	LD r,(HL)
LD A,(DE)	LD r,(IX+d)
LD A,(HL)	LD r,(IY+d)
LD A,(IX+d)	LD r,A
LD A,(IY+d)	LD r,B
LD A,(address)	LD r,C
LD A,A	LD r,D
LD A,B	LD r,E
LD A,C	LD r,H
LD A,D	LD r,L
LD A,E	LD r,32 (any number)
LD A,H	
LD A,I	
LD A,L	
LD A,32 (any number)	
LD A,R	
LD (address),A	

I think this should make it clear why one can sometimes use A to LD where
no other register would suffice. Note some of the strange things registers
can be loaded with - like (IX+d) and R and I. Don't worry, it's nothing
you need to know. These are better tools in the hands of an experienced
programmer.

9

By the way - the little r stands for any register, e.g. A,B,C,D,E,H or L.
Remember to include an ORG instruction at the beginning of your assembly
programs: I may omit them from time to time. This means you can make
the ORG value anything you like.

Now, before I continue with new material: some people have asked me what
the use of machine code is. I repeat: speed! NO other computer language
on the Spectrum will execute as fast as machine code. Also, it saves
a great amount of space - a 10K Basic program may take up a mere 2K if
written in machine code. I sent in a game listing to the newsletter
written in pure machine code (I do not know if they would have published
it!) involving you having a dogfight with five other sprites and a
protective shield acting as a barrier for both sides. Apart from the
fact that everything is pixel-smooth movement, the entire listing fits
into 1,8K. Try doing that in Basic. So stop looking depressed and read on.

*Variables

Instead of using variables like in Basic, we can use addresses in machine
code if the registers are not enough. We will come to this later, but
I would like to point out that because a program starts at address 40000
it doesn't necessarily mean that you would use RANDOMIZE USR 40000 to
RUN it: the first byte may be a variable, or even the first few bytes.
The program may only start at 40100. I shall, however, avoid giving
programs like this and place my variables at the end. When writing you
own listings, though, be alert: you may have placed some variables
after ORG causing the program to crash when RANDOMIZE USR'd incorrectly.
Keep this in mind, but don't worry - yet.

*Jumping and calling

In Basic, we have some useful JUMP commands. These are GO TO and GO SUB.
If, for instance, we wished to jump to line 20 without returning if
a variable B is greater than 10, we would use

IF B>10 THEN GO TO 20

Now, this is a very popular way of making decisions on a computer. In
Basic, syntax such as

IF B=2 THEN LET A=5

is allowable, whereas in machine code it is not: you MUST do a jump called
a conditional jump. How this is done comes up later. For the moment, look
at the following:

JP address
JR displacement
CALL address

JP and JR are the machine code equivalents of GOTO and CALL the equivalent
of GOSUB. Consider, however, that assembly programs are not dependant on
line numbers: JP 10 means something completely different than GOTO 10.
Let us look at JP first.

JP is probably the simplest of the three. It means, simply, JumP! If I
said JP 40000, I am instructing our friend the CPU to jump from where it
is currently to address 40000 and start running the instruction it
encounters there.

This could be a bit akward. Say you wanted to jump to a place in your

program where register A is loaded with the content of register B, but
you do not know where (e.g. at which address) the instruction is stored.
Luckily, assemblers have things called LABELS: it is like a line number,
but written as a word. Instead of JP 60000, we could use JP MARKER. Our
hypothetical program would look like this using a label:

```
          ORG 50000
          JP MARKER
          RET
MARKER LD A,B
          RET
```

Do you follow? I set a label called MARKER in the label field to the
extreme left, started the program at address 50000 and then JUMPed over
a RET instruction to the label MARKER, where I loaded A with the content
of B and RETired back to Basic. If I wrote

```
          ORG 50000
          JP MARKER
          RET
          LD A,B
MARKER RET
```

the program would start, JUMP to MARKER, skipping the RET instruction
and the LD A,B instruction and RETire to Basic.

Consider the first example again and then look at this program:

```
          ORG 50000
          JP 50004
          RET
          LD A,B
          RET
```

It would have EXACTLY THE SAME EFFECT, since the LD A,B instruction is
situated, in fact, at address 50004. But why do the calculation if the
assembler can do it for you?

When the JP is converted to numbers, it takes up 3 bytes. The first is
a byte telling the CPU to JUMP. The second is the first bit or LOW BYTE
of the address where it has to jump to and the third is the second bit
or HIGH BYTE of the address where it has to jump to. There is another
instruction which takes up only two bytes of memory and can sometimes
be used as a substitute for the JP instruction: this is the JR or
Jump Relative command. It actually tells the computer to jump up to 175
bytes forward or backwards from a certain point and can be used exactly
like JP.

```
          ORG 50000
          JR MARKER
          RET
MARKER LD A,B
          RET
```

Note, however, that if MARKER were more than 175 bytes away from the JR
instruction, the program would not work.

We now move on to CALL. CALL is used in the same way as JP, but expects a
RET - like the GOSUB and RETURN combination in Basic. Consider this prograr

```
          ORG 50000
          CALL MARKER
```

```
MARKER   LD A,B
         RET
```

The program starts at 50000 and CALLs a subroutine called MARKER. MARKER
LDs A with the content of B and then RETurns to the instruction after
the CALL - that is, the first RET which makes the program RETurn to Basic.
It is now clear that the RET instruction has two uses: either to RETurn
from a subroutine without going back to Basic, or to end the program.

*INC and DEC

These are quite useful commands in machine code. They can be used to
either ADD one to a number in a register or to SUBTRACT one from a
number in a register.

```
LD A,10
INC A
RET
```

This program loads A with 10, increments it and RETires. The value of
A is now 11. We could have used DEC A, and the value of A would be 9
when the routine RETires.

Any of the A,B,C,D,E,H or L registers can be INCremented or DECreased in
this way.

There is also one REGISTER PAIR which can be INCremented or DECreased:
the HL register pair. Remember: like the A is a favoured register, the
HL register pair is favoured, e.g. you can do more with HL than other
register pairs.

*ADD and SUB

If, however, we wish to add or subtract more than one from the value in
a register, we use ADD and SUB. There are also the ADC and SBC commands,
but we will use these when I explain flags in machine code. Unfortunately,
it is possible to ADD and SUBtract from ONLY THE FAVOURED A REGISTER.

```
LD B,10
LD A,B
ADD 10
LD B,A
RET
```

This program demonstrates how you would add 10 to another register.
Firstly, you load A with the other register's content, then you do the
ADDing and than you load the new value in A back to B.

Some assemblers need to be instructed ADD A,10 instead of just ADD 10.
The HL register pair can be used to ADD numbers bigger than 255. It
CANNOT BE USED TO SUBTRACT. To add 16384 to 16384, the following
program would be correct:

```
LD HL,16384
LD BC,16384
ADD HL,BC
RET
```

The answer will be stored in HL. Note that we cannot address HL directly
when we add. We have to use ADD HL,BC or ADD HL,DE. ADD HL,16384 is not
acceptable. This is not the case with single registers. Here are all the

SUB and ADD possibilities:

ADD (HL)	SUB (HL)
ADD r	SUB r
ADD 32 (any number)	SUB 32 (any number)
ADD HL,rr	

r stands for any register, rr for any register pair. Till next month...

PLOT? command for the QL
by Hans Loedolff

QL users would be aware the single graphics command that the QL lacks. The command is used to retrieve the colour value of a pixel on the screen.

I added the command as a function. It uses the pixel coordinate system to define the positions. The program works only in mode 4.

The first program uses the top left corner of the screen for the 0,0 point, while adding the variation, the top left corner of the specified window would be used as the 0,0 point.

FRINT PLOT(100,100)

would return the value of the colour 100 pixels across and down from the top left corner of the screen.

If the variation is added, the window number should be added.

eg: FRINT PLOT(1,100,100)

The first parameter represents the channel. Their is no channel default and an error would occur if it is not specified.

Program 1

```
28000 DEFine FuNction plot(x,y)
28005  LOCal v,h,c
28010  IF x>511 OR y>255 OR x<0 OR y<0
       :RETurn 0
28020  v=PEEK_W(131072+(x DIV 8)*2+128*y)
28040  h=2^(7-x MOD 8)
28050  IF (h && v)=h:c=2:ELSE c=0
28060  h=2^(15-x MOD 8)
28070  IF h=32768 :h=-32768
28080  IF (h && v)=h :c=c+4
28090  RETurn c
28100 END DEFine plot
```

Variation (add the following lines to the above program)

```
28000 DEFine FuNction plot(ch,tx,ty)
28002  LOCal v,h,c,x,y,p :x=tx :y=ty
28005  p=PEEK_L(PEEK_L(163960)+4*ch)
28010  IF x>PEEK_W(p+28) OR y>PEEK_W(p+30) OR
       x<0 OR y<0:RETurn 0
28015  x=x+PEEK_W(p+24) :y=y+PEEK_W(p+26)
```

The following programs is only two of a thousands examples of what can easily be achieved through the new command. The programs use the variation version.

The first program prints enlarged copies of the text in the top left corner of #1. You can play around with the numbers for interesting effects.

```
10 OPEN #5,scr_512x256a0x0
20 AT 0,0:PRINT 'Quantum Leap'
30 FOR y=0 TO 10
40  FOR x=0 TO 72
50   BLOCK #5,6,20,x*6+20,y*20+10,plot(1,x,y)
60  END FOR x
70 END FOR y
```

The following program copies the top left corner of #1 sidewhays to a space just below it. Frint text in the space before runing the program.

```
10 FOR y=0 to 50
20  FOR x=0 to 50
30   BLOCK 1,1,y,100-x,plot(1,x,y)
40  END FOR x
50 END FOR y
```

Try writing a program that copies upside-down!

13

TASCOPY PC
THE GRAPHICS PROCESSOR
© Tasman Software Limited 1987

TASWORD PC
THE WORD PROCESSOR
© Tasman Software Limited 1986

TASCOPY PC
THE GRAPHICS PROCESSOR
© Tasman Software Limited 1987

TAS-SPELL PC
THE SPELLING CHECKER
© 1987 Tasman Software Limited and Longman Group (UK) Ltd

TASPRINT PC
THE STYLE WRITER
© Tasman Software Limited 1987

Discover the full potential of your Amstrad PC with... and others!

FULLY IBM Compatible

EASY to use

COMPLETE instruction manual

ONLY R89.00 each, P&P 1.50 each

BUY FIVE for only R84.00 each, P&P 3.50 inc!

We also stock SOFTCOVER titles!

Call for more details!

OR   Send a SAE + R1

Some titles:-	
VP PLANNER	R142.00
VP INFO	142.00
TURBO CAD	142.00
BOOKMARK	70.99

Since announcing that we intend stocking this PC we have had tremendous response. this is the best value for money EGA/CEA/HERCULES system around and we have won over even some skeptics who were taking the Taiwanese route.

They've been impressed by the attractive appearance, quality and manufacture of this system - so join them in your quest for the optimum deal:-

* 640K standard
* 8086 processor chip
* mouse and joystick adaptor
* socket for 8086 Maths co-processor
* serial and parallel ports for printers
* 3 extra slots
* 4 system disks, incl. GEM Paint, GEM Desktop, and MS DOS 3.2

You can get all this and more from only R2999.00!

AMSTRAD is backed by MAYNARDS, supplied by

AMABOP INDUSTRIES & INFORMATION ENTERPRISES.

HARD DRIVES

	RRP	Our Price
Seagate drive and Western digital controller and cables	950.00	719.00
P&P	10.00	

FLOPPY DRIVES

	RRP	Our Price
360K JVC 5.25" DS/DD	299.00	199.00
P&P	6.00	

LIMITED SPECIAL OFFER

14

Tasman
PERFECTION IN PROFESSIONAL SOFTWARE

## STOCKTAKING IN FULL SWING

THE LANGUAGE OF COMPUTING   by Jacqueline Snell

LIKE ANY OTHER AREA OF SPECIALIST KNOWLEDGE, COMPUTING HAS ITS OWN LANGUAGE TO EXPLAIN HOW THE COMPUTER WORKS. TO THE UNINITIATED, THESE CONCEPTS SEEM BAFFLING. THE COMPUTER UNDERSTANDS NOTHING BUT NUMBERS - BUT THEY'RE NUMBERS WITH AN UNFAMILIAR LOOK ABOUT THEM.

BITS AND BYTES

The words 'bits' and 'bytes' are used whenever computers are written about. They are terms that describe the way computers store and use numbers. Computers, and for all their mathematical wizardry , use only two numbers - zero and one. Bits and bytes are ways of representing combinations of these two numbers. A bit (the word is derived from BInary digiT) is the smallest unit of mathematical information a computer can handle. It is the computer's way of representing the two numbers zero and one. THOUGHT OF IN ANOTHER WAY, a BIT CAN COUNT, but only from ONE TO ZERO. Picture a board with just a single hole drilled into it, which can be filled with a peg. Even though it is a single hole, it can represent either zero (the hole has no peg in it) or a one (the hole does have a peg in it). A computer does the same with an electrical signal that is either off (to represent 0) or on (to represent 1). If the board has two holes, it can show four possible combinations of holes and pegs, ie COUNT FROM 0 TO 3. If eight holes are used there are 256 different combinations. Computer's use groups of EIGHT BITS and such a group is called a BYTE. Each byte can represent a number ranging from 0 to 255 decimal and 11111111 binary. (Do you remember how to use your fingers to convert binary to decimal? Refer to April's newsletter.) Two bytes can be grouped together to make what is technically called a WORD. A WORD can be expressed using 16 BITS or four HEX digits, and represent a number from 0 to 65535 decimal; 1111111111111111 binary and FFFF hexadecimal.


Pulse   No-pulse

COUNTING IN BINARY

Counting in binary

8s   4s   1s

2s

1000 s   10 s   1 s

100 s
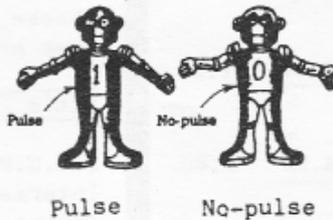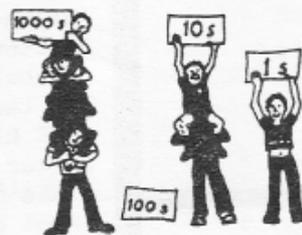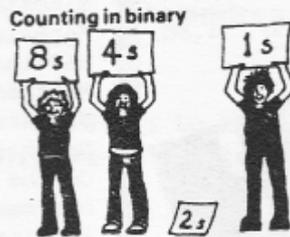
A PIECE OF WOOD WITH ONE HOLE INTO WHICH A PEG MAY BE PLACED:
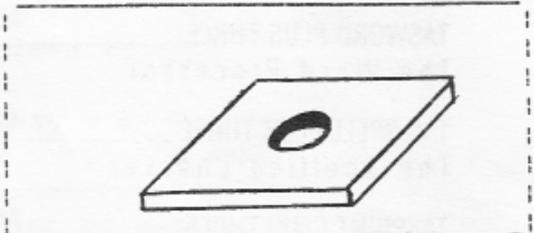
This board can represent a 0 or a 1; i.e. it can count from 0 to 1 in binary and decimal.

A PIECE OF WOOD CONTAINING TWO HOLES INTO WHICH PEGS MAY BE PUT:
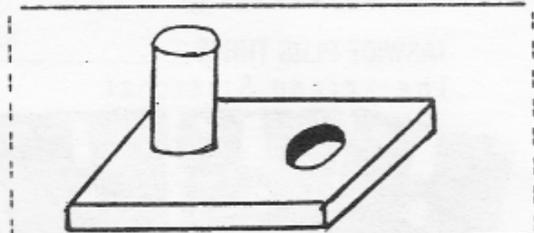

This board can represent :
1. 0 and 0 binary and decimal
2. 0 and 1 binary and decimal
3. 1 and 0 bin and 2+0=2 dec.
4. 1 and 1 bin and 2+1=3 dec.
i.e. it can count from 0 to 3 decimal or 0 to 1 binary.

A PIECE OF WOOD CONTAINING EIGHT HOLES INTO WHICH PEGS MAY BE PUT:
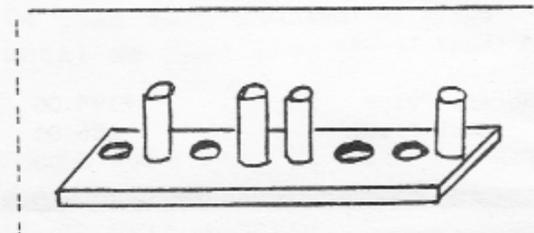
This board can represent 256 combinations, states or permutations of pegs and holes, 1s are represented by pegs and 0s by holes. In other words it can count from 1 to 255 decimal=1+2+4+8+16+32+64+128=255; 11111111 Binary.
                                P.T.O.

One bit →
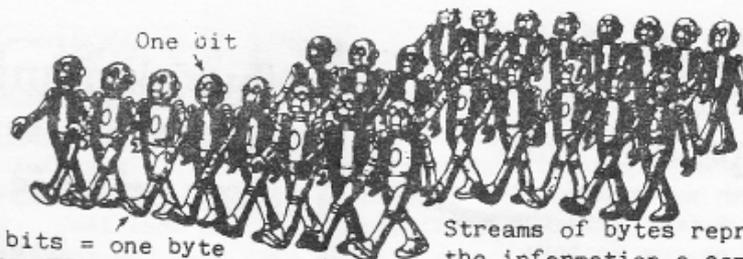
Eight bits = one byte

SOMETHING FOR YOU TO DO:
=========================

Do you know how to convert binary to hexadecimal? Well, if not this is something new for you to learn and to show your teacher and friends!!

In the hexadecimal number System, 0 to 9 is represented in exactly the same way as we do in our decimal number system; then from there on there are a few differences:

Decimals	Hexadecimals
========	============
10	A
11	B
12	C
13	D
14	E
15	F

The decimal number system is based on 10; the binary number system is based on 2 and the hexadecimal system is based on 16.

Now, convert the undermentioned binary four bits into hexadecimal:

1111 1111 1111 1111

Answer:
Firstly, convert to decimal: each four bit represents from left to right, 8+4+2+1=15. Then, convert to hexadecimal and each 15 in decimal notation will be represented by an 'F' in hexadecimal notation. So finally, 1111 1111 1111 1111 BIN is represented by FFFFh. If we are using the hex notation we write 'h' at the end of the number.

A LISTING FOR YOUR COMPUTER TO CONVERT DECIMAL TO HEXADECIMAL NUMBERS:

```
10 REM **PROGRAM FOR CONVERTING
   DECIMAL TO HEXADECIMAL**
20 BORDER 0:PAPER 1: INK 7
30 PRINT "Enter decimal  number
   to be converted?"
40 PRINT ""
50 PRINT "DECIMAL NO:","HEX NO:"

60 PRINT "========== ","======"
70 INPUT n
80 DEF FN D$(n,d)="0123456789AB
   CDEF"(INT(n/16^d)-16*
   INT(n/16^(d+1))+)
```

Streams of bytes represent all the information a computer uses.

```
90  LET A$="":FOR d=LEN STR$
    n-1 TO 0 STEP -1
100 LET A$=A$+FN D$(n,d):
    NEXT d
110 IF A$(1)="0" THEN LET A$=A$
    (2 TO):GOTO 120
120 PRINT n,A$
130 GOTO 70
```

LISTING TO CONVERT HEX NOS. TO DECIMAL NUMBERS

```
10 REM ***PROGRAM  TO  CONVERT
   HEX NOS INTO DECIMAL NOS***
20 BORDER 0:PAPER 1:INK 7
30 PRINT "Enter hex number  to
   be converted?"
40 PRINT "HEX NO","DECIMAL NO"
45 PRINT "======","=========="
50 INPUT H$
55 LET value=1:LET dec=0
60 FOR h=LEN h$ TO 1 STEP -1
65 LET dec=dec+(CODE h$(h)-
   48-(7 AND h$(h)>"9")-
   (32 AND h$(h)>="a"))*value
70 LET value=value*16
75 NEXT h
80 PRINT h$,dec
90 GOTO 50
```

NB. The symbol after the digit === 16 in line 80 in first listing is obtained by pressing symbol shift + H.

To brush up on binary learnt in April's newsletter convert FFFFh, 1111 1111 1111 1111b to decimal format 65535!

	1111	1111	1111	1111	
Groups:	4	3	2	1	
1:	1+	2+	4+	8=	15
2:	16+	32+	64+	128=	240
3:	256+	512+	1024+	2048=	3840
4:	4096+	8192+	16384+	32768=	61440

Although computers use a pure binary system, humans often write the numbers stored inside a computer using HEX notation - after all, the number 3AF0h is far likely to be easily and correctly read than 0011101011110000 in sixteen bit binary notation. Your Spectrum has a keyword 'BIN' which provides a means of entering in BINary. PRINT BIN 100 = 4 dec (4+0+0); PRINT BIN 111 = 7 dec (4+2+1) and finally PRINT BIN 11111111 = 255 dec (128+64+32+16+8+4+2+1).

18

NEWS

Cambridge-based MILES GORDON TECHNOLOGY plans to launch a Spectrum compatible machine towards the end of this year. Developments have been going on for some months and industry speculation that the firm would be releasing a machine have been doing th rounds almost since the Amstrad purchase of the Sinclair Research rights.

According the a spokesman at MGT, the machine is superior to anything that came out of the Spectrum.

MGT has high hopes for SAM, the computers working name. The firm is eyeing the Third World for the possibility of potentially lucrative licensing aggreements. Apparently the machine is designed specifically to be built easily anywhere in the world.

Also, it has been pitched as a "mass market product" and one "ideal for the education market". Spectrum compatibility is only one of the modes SAM runs under but for the moment, that would appear to be the most significant aspect of the machine!

ATTENTION Z88 OWNERS!!!

Here is some very exciting news for you!

N. van der Merwe se navraag oor hoe om "graphics" oor die skerm te laat "loop" (Mrt. 88): die antwoord lê waarskynlik opgesluit in 'n masjienkode-roetine wat die skets (mannetjie of wat ookal) "pixel"-gewys oor die skerm laat beweeg, moontlik deur gebruik te maak van 'n sg. "interupt-driven" roetine. As die figuur letterlik moet LOOP, m.a.w. as hy sy bene moet beweeg sal 'n AANTAL verskillende "graphics benodig word, wat elkeen 'n DEEL van die bewegeng voorstel. Die vorige skets word gewoonlik uitgevee met "PRINT; OVER 1..." en 'n HERHALING van die skets, of met "PRINT AT..." (die huidige posisie van die skets); " " (twee aanhalingstekens met 'n spasie tussenin). In BASIC word die "graphics" dan aangeskuif deur die tweede koördinaat (posisie) van die skets telkens met een te vermeerder.

Probeer die volgende:
```
10 LET X = 10:  LET Y = 0
20 PRINT AT X,Y; "A": PAUSE 10: PRINT AT X,Y;" "
30 IF Y 31 THEN LET Y=Y+1: GO TO 20
```
Vervan "A" hierbo met jou eie UDG

NOTICE BOARD

Subscription News

NAME _____ _____ TEL _____ MEMBER REF #:_

ADDRESS _____

To: INFORMATION ENTERPRISES, PO Box 35644, Menlo Park, 0102

DESCRIPTION   PRICE
###

_____   R_____.___

_____   R_____.___

_____   R_____.___

_____   R_____.___

_____   R_____.___

_____   R_____.___

===

SUBTOTAL: R_____.___

POSTAGE: R_____.___

TOTAL: R_____.___

	Surface Mail	Air Mail
SOFTWARE		
1 Tape	1.15	1.50
2 Tapes	1.50	1.85
3 or more tapes	2.30	2.80
Packs	1.85	2.30
HARDWARE		
Interfaces (all)	1.85	2.30
Disc Drive (single)	5.90	9.00
Disc Drive (double)	7.00	11.50
Expansion pack	3.45	5.75
A4 Printer	10.00	13.80
Timex Printer	4.45	5.90
Joystick	1.80	2.30
Monitor (green)	8.00	10.50
QL computer	8.00	14.50
Spectrum +	8.00	14.50
Spectrum + 2	10.00	13.80
Spectrum + 3	11.90	14.50
PARTS		
Membrane	2.30	2.80
All other parts	1.85	2.00
STATIONARY		
Box of paper (2000)	3.00	5.00
Alphacom paper (1)	1.00	1.50
Alphacom paper (2)	1.50	2.00
Alphacom paper (5)	3.00	4.00
A4 roll of paper	2.80	4.90
KEYBOARDS		
All keyboards	1.85	3.45
Z88 computer	5.00	7.50