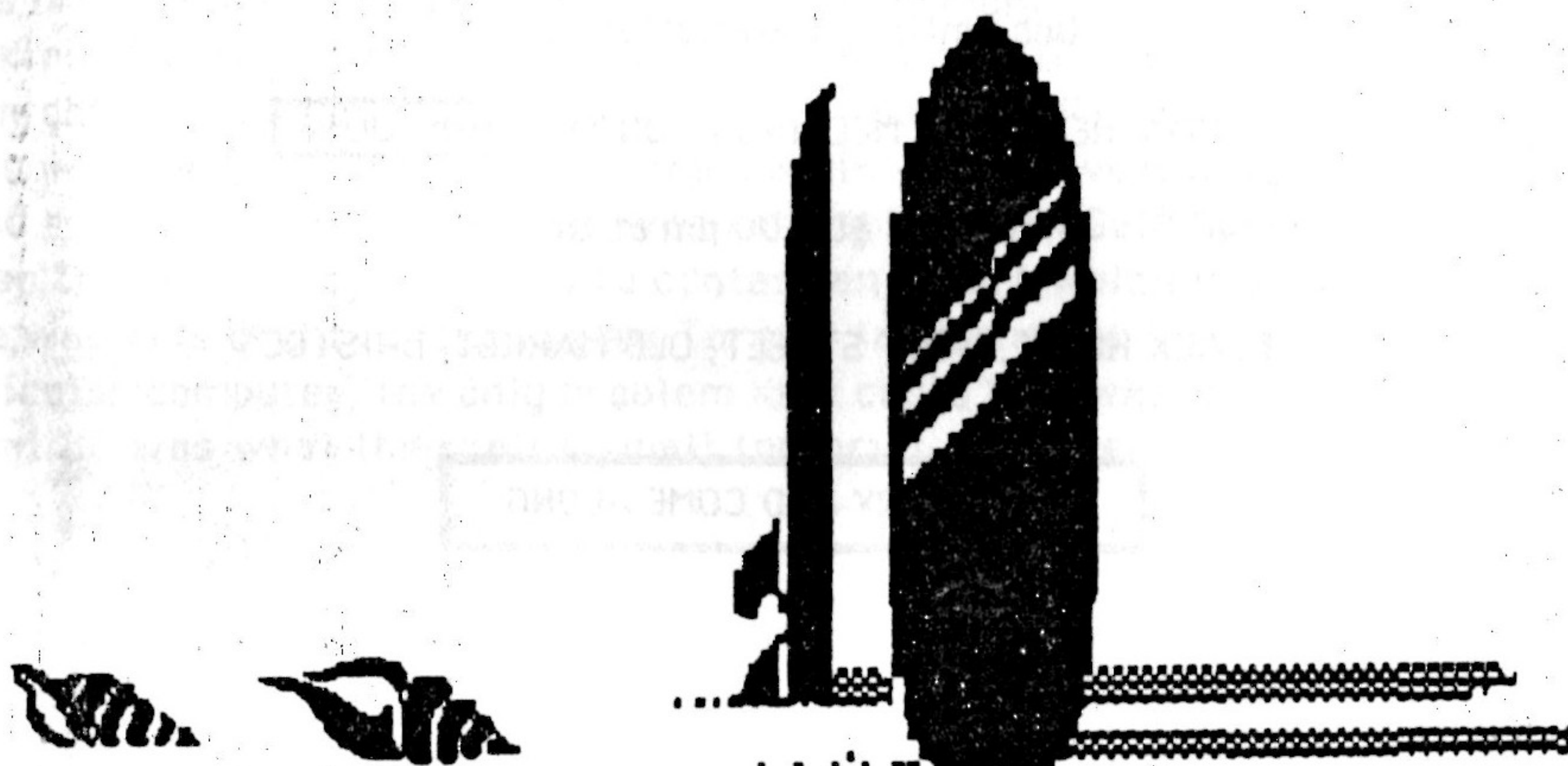
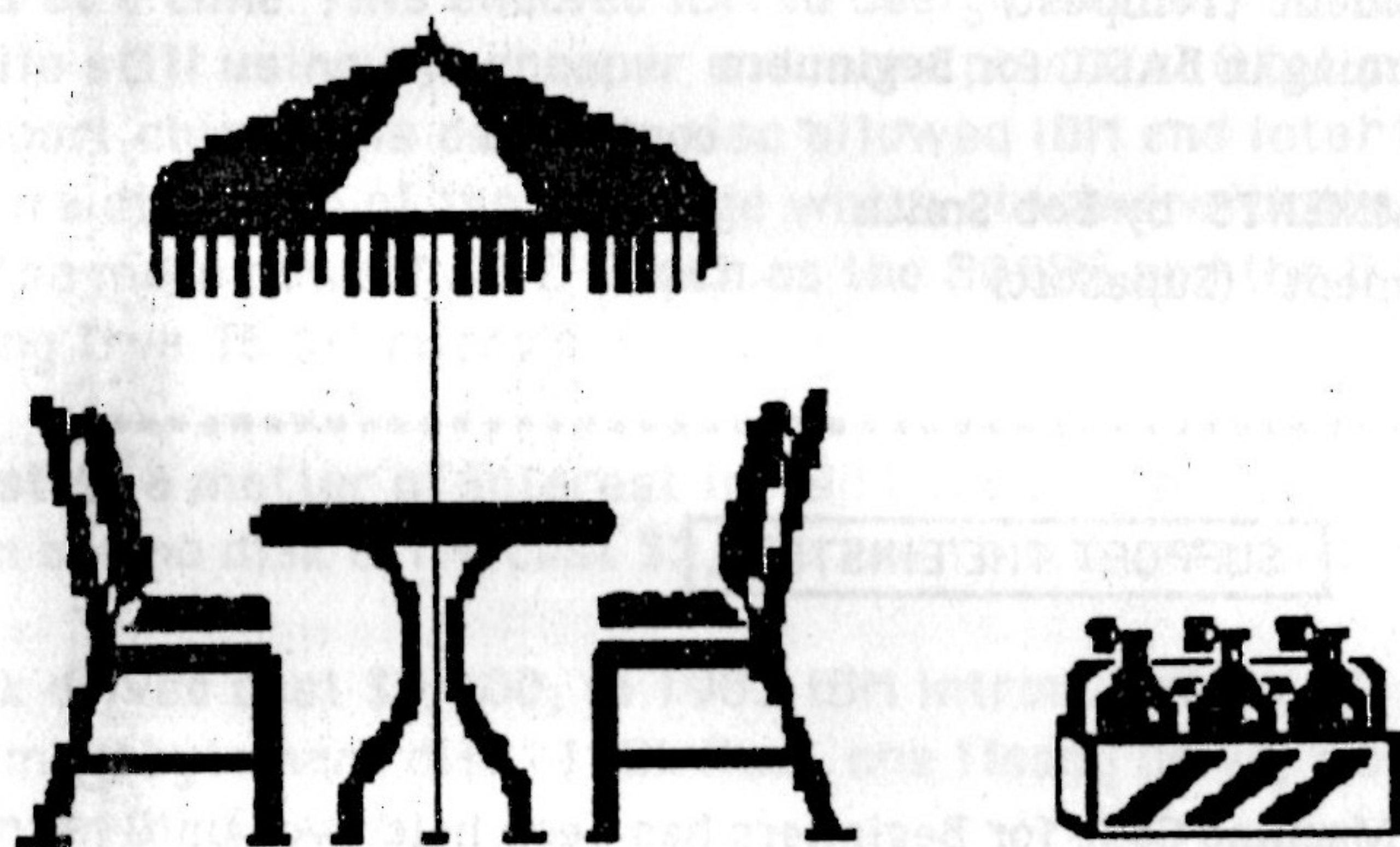


AVON NEWS

VOL.3.NO.5.



PAGE INDEX

- (1).....= 16 BIT CORNER by Bob Smith
- (2).....= " " continued
- (3).....= " " continued
- (4).....= " " continued
- (5).....= LETTER from the North
- (6).....= SUMMARY OF MICROSOFT BASIC COMMANDS, Functions & Operators
- (7).....= " " continued
- (8).....= " " continued
- (9).....= " " continued
- (10).....= " " continued
- (11).....= " " continued
- (12).....= Advertisment (Tempest)
- (13).....= Programming in BASIC for Beginners
- (14).....= " " continued
- (15).....= CLUB COMMENTS by Bob Smith
- (16).....= Advertisment (SupaSoft)

SUPPORT THE EINSTEIN

Editors Note.....

Johns article on Machine Code for Beginners has been held over until next month as he will be on holiday.

Bob Smith Newsletter Editor

THE NEXT CLUB MEETING IS ON THE 8th JUNE

starting at 8.00 pm at the

BLACK HORSE, WEST STREET, OLD MARKET, BRISTOL

PLEASE TRY AND COME ALONG

THE 16 BIT CORNER

by
Bob Smith



Well I have done it at last !, as you know I have upgraded to an IBM compatible XT.

This probable means very little to most of you so I will try and expand on this. In 1979 IBM decided to bring out a computer which was going to be different than the 8 bit micro's they were already producing. They came up with the solution of using a microprocessor which was being produced by Intel called the 8088, this microprocessor was internally a 16 bit chip but it communicated with other circuitry by using only 8 bits at a time. This enabled IBM to design and build a 16 bit computer while still using the cheaper and more plentiful 8 bit processors as support chips, this decision also allowed IBM and Intel to define the future direction of the PC range which started with the 8086 processor to the more powerful PC's such as the 80286 and the 80386 the latter being true 16 bit micro's.

Just as a matter of interest in 1981 the original basic IBM PC with 64k ram and no disk drive cost \$2,000 and fitted with two single-sided 160k

disk drives cost \$3,000, in 1983 IBM introduced the XT version with a 10 megabyte hard disk, 128k Ram, one floppy drive, mono monitor and printer for \$5,000.

This brings me back to my computer, for some months now I have been thinking of changing my Einstein for an IBM compatible so every fortnight I have been getting the magazine "MICROMART" and looking through it for that "once in a lifetime bargain", this I eventually found !! A gentleman in Swindon was selling his DeltaGold Sprint 88 XT, so the first thing I did was to contact an expert which in this case was Kris Brunt of Maplins (ex Target) fame and ask him about this particular computer, the only problem Kris could find was that the Sprint 88 was what they call a small footprint computer which

measured only 260mm X 260mm X 80mm but this could be got over by taking the computer apart and fitting it in another case so it was of to Swindon to buy it.

After getting back to Weston I was like a child with a new toy I couldnt wait to get it fired up, this is where my problems started because IBM's use a disk operating system called MS-DOS which was designed for the IBM by Microsoft, this DOS loads when the computer powers up just like the Einstein's XTAL DOS but this is where the similarity ends !. The MS-DOS I had was configured for an USA keyboard so therefore the £ was a \$ so I got on the phone to Mike Maddock to ask his advice, this is when I found out about two files which are called AUTOEXEC.BAT and CONFIG.SYS, these two files have to be setup by you so that when you boot-up a file called COMMAND.COM looks for the AUTOEXEC file and that in turn looks at the CONFIG file and if you have written them properly the computer will be have been setup to your personal requirements.

I have setup my Autoexec and Config files as follow :-

AUTOEXEC.BAT *****

echo off (turns on or off the display of batch commands and displays a message on the screen)

path=c:\bin;c: (set directory search order)

keybuk (enable foreign language keys)

gmouse (mouse system driver)

timepark 1 (a public domain program which parks the hard disk when the drive is not in use)

prompt \$p\$g (set the system prompt)

echo Smith Systems !!! (my bit of vanity)

CONFIG.SYS *****

buffers=20 (sets the number of disk buffers)

device=c:\bin\sys\ansi.sys (assigns new or changed devices)

files=20 (sets the maximum number of files handles)

fcbs= 12,12 (file control block)

lastdrive=f (sets the last logical MS-DOS disk drives)

country=044 (sets the country that the computer is located)

I think the best thing I did was to buy a book called the "MS-DOS USER'S GUIDE" which I recommend that you get because you will find you will need it to understand the MS-DOS operating system !!!.

Well that was that I had my computer speaking to me in English and not American so the next problem was to remove the innards and put it in a larger case. The reason for this is that the original computer had a 3.5" drive fitted in it and an external 5.25" drive so the idea was to have the two drives fitted inside the case with enough space to add a hard disk and to increase the slots from one to four which would also allow me to fit a EGA card and monitor at a later date.!!!!

At the Einstein Show I had a chance to talk to Tony Brewer who advised me to go to a company at Staverton near Cheltenham so it was of to Staverton to find a company called CONTINENTAL Ltd which are located at Staverton Technology Park, this company puts computers together and they quickly found a suitable case for me with a 180 watt power supply and two spaces for 3.5" drives and one 5.25" drive also a Bus Bar which turns my original 1 slot into 4 slots.

Back in Weston I took my original computer apart and with the help once

again of Kris Brunt and fitted it all into the new case. The next job was to find someone who wanted my Olivetti Daisy-Wheel printer for cash or would swap it for a hard-disk and once again turning to the pages of "MICROMART" I found a company called EBL at Staple Hill in Bristol who were quite happy to exchange my printer for a hard-disk and controller card.

So here I am I, the proud owner of a 8088-1, 10 mhz computer with a 5.25" 360k drive, 3.5" 720k drive and a 21 meg hard-disk and I can expand it at a later date with a EGA card and Colour Monitor.



If you have upgraded to a 16 bit computer lately then let us know about how you are getting on with it, perhaps you can do a short article on how you are getting on or what programs you run on it ?????.

I would like to thank the company's and people that have helped me get my computer together and they are :-

My Wife (for lending me the money)

CONTINENTAL LTD,
STAVERTON TECHNOLOGY PARK,
CHELTENHAM,
GL51 2DE

Tel. No. CHURCHDOWN (0452) 856798. TELEX: 265671 REF:WJJ 032

Gary Drake,
E.B.L. (Equipment Brokers Limited)
E.O.S. HOUSE,
145 STAPLE HILL ROAD,
FISHPONDS,
BRISTOL.

BS16 5AG
Tel. No. 0272/701544

KRIS BRUNT of MAPLINS Gloucester Road (Honorary A.E.U.G. Member)
MIKE MADDOCK (A.E.U.G. Member)

16, Heath Close
BOLTON
BL3 3RW

17th May 1989

Dear members,

I was very sorry to miss the Einstein show at Birmingham and the chance to renew acquaintances made last November. However, I thought that it was long past time that I put finger to keyboard to send this contribution to the newsletter.

Living so far from Bristol makes me wonder about the geographical spread of members. Are there many other far flung parts of the club?, and if so, could I suggest that we use these columns to introduce ourselves?

My name is Steve Ryder and I live in Bolton which is about twelve miles from Manchester just on the edge of the west Pennine moors. I bought my Einstein from Dixons when they were being sold off as discontinued stock and have been very satisfied with the machine. B&H supplied a second 3" drive and made a good machine better. The only fault which occurred was the -12V supply failed!, but the replacement of the regulator chip in the power supply solved this at the cost of only 32 pence.

I use the machine for general interest and the kids play games on it using a Spectrum type joystick and an adaptor built from a circuit provided in a copy of the UKEUG magazine. This works fairly well except that there is a gradual downward drift probably due to my not using very high tolerance components.

Is there anyone who can help me with the customisation of DBASE2 for the Einstein?. I have transferred a copy bought for a different machine and despite the assistance of a couple of correspondents via the bulletin board I still can't get the screen format right.

Anyway, enough of my maunderings, I hope to see some of you in the future.

Cheerio.

Steve Ryder
Membership no.060

Summary of Microsoft BASIC commands, functions & operators

Notation used in this document

(##) command accepted by interpreter but not by compiler
 <...> parts of command enclosed thus are optional
 ... indicates a list of similar items separated by commas
 =NAME indicates that NAME is a function
 char abbreviation for "character" (letter, digit or symbol)
 ln file number (in the range 1-15) specified by OPEN
 x an expression (e.g. A + LOG(B))
 i an integer expression
 x\$ a string expression (e.g. AS + RIGHTS(BS,4))
 var a variable name

Running BASIC

After a CP/M prompt, simply type the word "BASIC", e.g.
 A>BASIC

There are several options which may be given on the command line after the word "BASIC". These are as follows:

filename name of a .BAS file to be run immediately
 /F:number of files max files to be open at once (default 3)
 /M:highest address highest memory address to be used by BASIC
 /S:max reclen max random-access record length (default 128)

Don't separate options by commas: the "/" is the separator.

Editing commands (##)

AUTO <ln,inc> auto line numbering from ln in steps of inc
 DELETE ln1 <- ln2> delete line ln1 <lines ln1 to ln2>
 EDIT ln enter EDIT mode at line number ln
 within EDIT mode:
 the SPACE bar leaves the current char unchanged
 the RETURN key leaves the rest of the line unchanged and exits the EDIT mode
 other special key effects:
 L : list line and put cursor at start
 I : insert char's (end with ESC)
 X : move to end of line & insert char's
 D : delete next char
 H : delete all following char's & insert
 Sc : move to next occurrence of char c
 Kc : delete all char's up to char c
 Cc : change next char to char c
 E : exit EDIT
 Q : exit EDIT, cancelling changes made
 A : cancel changes, list original line
 LIST ln1 <- ln2> list line ln1 <lines ln1 to ln2> on screen
 LLIST ln1 <- ln2> list line ln1 <lines ln1 to ln2> on printer
 MERGE "file" merge file into current program
 NEW close all files, discard contents of memory
 RENUM <new,old,inc> renumber program lines (default: 10,1,10)

Operating commands

CONT continue program after ^C, STOP or END (##)
 FILES <"file set"> list file names <may use (e.g.) "*.com">
 LOAD "file" <,R> load SAVED file <& RUN keeping data files> (##)
 KILL "file" delete file from disk
 REM <comment>
 RESET use after changing disk, to read new directory
 RUN <ln> run current program starting at line ln
 RUN "file" <,R> load file & run <keeping old data files open>
 SAVE "file" <,A> <,P> save on disk <ASCII format> or <protected> (##)
 (use ASCII format if you want the file to be readable outside BASIC, e.g. by a text editor)
 SYSTEM leave BASIC and return to CP/M

Declaration commands

DATA const,... store constants to be read by READ
 DEFINT range declare as integer
 e.g. DEFINT I - N declares the default type to be integer for variable names starting with letters I - N
 DEFSNG range declare as single-precision floating-point
 DEFDBL range declare as double-precision floating-point
 DEFSTR range declare as string
 DEF FNname <(parameter,...)> = expression
 define function (all function names must start with FN)
 DIM var(),... specify maximum array subscripts
 OPTION BASE n min value for array subscripts: n = 0 (default) or 1

Variable types

A variable is assumed by default to be single-precision floating-point unless a) the first character of its name is mentioned in a DEF<type> statement, or b) the last character of its name is one of the following special characters:

Postfix	Example	Type
%	I%	integer
!	VAR!	single-precision floating-point
#	DVAR#	double-precision floating-point
\$	STR\$	string

Conversions between types are rounded where necessary.
 DEF<type> statements don't apply to variables typed by postfix.

Arithmetic, trigonometry & random numbers

x = y + z x becomes (y plus z)
 x = y - z x becomes (y minus z)
 x = y * z x becomes (y times z)
 x = y / z x becomes (y divided by z)
 x = y ^ z x becomes (y to the power of z)

The following two operators round off their operands to integer values and truncate the result to an integer value (if necessary):

i = j \ k i becomes (j divided by k)
 i = j MOD k i becomes (the remainder when j is divided by k)

=ABS(x) absolute value
 =ATN(x) arctangent in radians

= COS(x) cosine in radians
 = EXP(x) e to the power of x
 = LOG(x) natural logarithm of x
 RANDOMIZE <x> initialise random number generator
 = RND <(x)> random number between 0 & 1
 = SGN(x) sign of x
 = SIN(x) sine of x in radians
 = SQR(x) square root of x
 = TAN(x) tangent of x in radians

Logical operators

a = b true if a equals b
 a <> b true if a doesn't equal b
 a < b true if a is less than b
 a <= b true if a is less than or equal to b
 a > b true if a is greater than b
 a >= b true if a is greater than or equal to b
 NOT a true if a is false
 a AND b true if both a and b are true
 a OR b true if either a or b (or both) are true
 a XOR b true if either a or b (but NOT both) are true
 a EQV b true if both a and b are true, or if both are false
 a IMP b true unless a is true and b is false

String operations

s\$ = t\$ + u\$ s\$ becomes the concatenation of t\$ and u\$, e.g. if
 t\$ = "GO" and u\$ = "AWAY", s\$ becomes "GO AWAY"

String comparisons:

Logical operators may be used on strings.

One string is "greater" than another if:

1. It follows the other alphabetically.
2. It's the same as the other, but longer.

= INSTR(<i>, x\$, y\$) position of y\$ in x\$, starting from pos i
 = LEFT\$(x\$, i) leftmost i char's of x\$
 = LEN(x\$) length of x\$
 LSET var\$ = x\$ left-justify x\$ in var\$
 = MID\$(x\$, i, j) str of length j from x\$ starting with the ith
 MID\$(x\$, n, <m>) = y\$ replace char's in x\$, starting at position n,
 with m char's from y\$ (default: all of y\$)
 = RIGHT\$(x\$, i) rightmost i char's of x\$
 RSET var\$ = x\$ right-justify x\$ in var\$
 = SPACES(i) string of i spaces
 = STRING\$(i, j) string of i char's of ASCII value j
 = STRING\$(i, x\$) string of i char's equal to first char of x\$

Conversion functions

= ASC(x\$) ASCII value of first char
 = CHR\$(i) one-char string of ASCII value i
 = CINT(x) convert to integer by rounding to the nearest
 = FIX(x) convert to integer by truncation
 = INT(x) convert to integer by rounding down
 = STR\$(x) string representing value of x
 = VAL(x\$) value of a number which is represented in a string

= HEX\$(i) string representing hexadecimal value of i
 = OCT\$(i) string representing octal value of i

= CSBG(x) convert to single precision
 = CDBL(x) convert to double precision

Algorithm control commands

CHAIN "prog" run prog in place of current program
 CLEAR set all var's to 0, str's to null (##)
 COMMON var,... pass variables to a CHAINED prog (##)
 END close all files and terminate program
 ERROR int set ERR = int and ERL = line number

FOR var = i to j <step k> loop with counter
 NEXT end FOR loop

GOSUB ln call subroutine at line ln
 RETURN return from subroutine

GOTO ln go to line ln

IF logical expression THEN statements ELSE statements

ON ERROR GOTO ln user routine replaces standard error handling
 routine must end with RESUME

RESUME <ln> <NEXT> resume program execution after error

ON exp GOSUB ln,... if exp = n, GOSUB nth line number in list

ON exp GOTO ln,... if exp = n, GOTO nth line number in list

READ var,... initialise variables from DATA statement

RESTORE re-use all DATA statements from beginning

STOP terminate program without closing files

SWAP var, var exchange values of two var's of same type

TRON/TROFF turns program tracer on/off

WHILE exp loop while exp is true
 WEND end WHILE loop

File-handling commands

CLOSE close all open files
 CLOSE fn,... close file numbers fn,...
 = EOF(fn) true when end of file on file fn
 (set up test before input statements)
 = LOF(fn) no. of recs in last extent read/written
 NAME "old" AS "new" change file name
 OPEN m\$, fn, "name", <reclen> open file : m\$ = "I" (input),
 "O" (output), or "R" (random-access)
 reclen sets the record length for a random-access file. The
 default reclen is 128 bytes; this is also the maximum reclen
 unless you've specified /S: <max reclen> in the BASIC command
 line.

Sequential input/output commands

INPUT <prompt\$> var,... input from console (adds " ?" to prompt)
 INPUT <prompt\$> var,... input from console (without " ?")
 INPUT# fn, var,... input from file
 LINE INPUT <prompt\$> x\$ x\$ = whole line input from console
 LINE INPUT# fn, x\$ x\$ = whole line input from file

=INKEY\$ one-char string from console, no echo,
null if no input waiting
=INPUT\$(n <,fn>) string of n char's read from console
<or file number fn>, no echo
=LOC(fn) no. of 128-byte blocks read/written
=LPOS(x) pos of print head within printer buffer
=POS(i) current cursor position (leftmost = 0)
=SPC(i) prints i spaces (within PRINT/LPRINT)
=TAB(i) space to pos i (within PRINT/LPRINT)
WIDTH <LPRINT> int set line width for screen <printer>
WRITE exp,... output exp's to screen in free format
WRITE# fn, exp,... output exp's to seq file in free format
free format: items separated by commas
: strings in quotes

Formatted output commands

PRINT <USING f\$> exp;... output exp's to screen
PRINT# fn, <USING f\$> exp;... output exp's to seq file
LPRINT <USING f\$> exp;... output exp's to printer

Examples:

PRINT X;Y\$;Z outputs the items as they are, without delimiters
PRINT X,Y\$,Z outputs the items with 14-column tabs between them

If there's a comma or semi-colon after the last variable in the list,
the next PRINT command will continue on the same line; otherwise the
end of the PRINT command means the end of that output line.

A = 12.312 : B = 4.56678
PRINT USING "A IS ###.## AND B IS ###.###"; A; B
produces output in the following format:
A IS 12.3 AND B IS 4.567

Within f\$ all char's are output as literals except those defined as
having special meanings, as follows:

! output only the first char of a string variable
\n spaces\ output the first n+2 char's of a string variable
(add spaces on the right if the string is too short)
& output a string variable in full
output a digit of a numerical variable
+ output the sign of the number (before or after it)
- output -ve sign only, after number
.. fill leading spaces with asterisks
\$\$ put \$ sign immediately before number
\$\$ fill leading spaces with asterisks and \$ sign
~ if before the decimal point, puts commas in number
after number, provide space for exponential notation
(underscore): treat next char as literal
% output % sign if number is larger than format allows

Random input/output commands

FIELD fn,len1 AS s1\$,... allocate strings to fields in buffer
=LOC(fn) next record number to be used
GET fn <,recnum> read record from file into buffer
PUT fn <,rec> write record from buffer to file

LSET s1\$ = x\$ move string data into buffer (left justified)
RSET s1\$ = x\$ move string data into buffer (right justified)

=MKIS(i) convert integer to 2-byte string

=MKSS(x) convert single precision to 4-byte string
=MKDS(x) convert double precision to 8-byte string
=CVI(s1\$) convert 2-byte string to integer
=CDS(s1\$) convert 4-byte string to single precision number
=CVD(s1\$) convert 8-byte string to double precision number

a string specified in a FIELD statement is a special kind of
string, and should only be altered by the LSET or RSET
statements - not by INPUT or LET statements.

Reserved variables

ERL contains line number of error
ERR contains error code

Referencing ports & memory locations

CALL i call machine-language subroutine at address i
DEF USRd = i declare machine-language function at address i (d=0-9)
=FRE(0) free bytes in memory
=FRE(x\$) free bytes in string space
=INP(pn) byte read from port pn
OUT ij send byte j to port i
=PEEK(i) byte read from memory location i
POKE ij store byte j in memory location i
=VARPTR(var) address of first data byte of var
=VARPTR(#fn) start address of buffer assigned to file fn
WAIT.pn,ij wait until (port pn XOR j AND i) nonzero

Miscellaneous

1. A series of statements may be written on one line, separated by colons (:). This is the way multiple statements may be written as clauses in IF statements.
2. Hexadecimal constants may be written using the prefix &H. Octal constants use the prefix &O, or just &.
3. A line in a BASIC program may be up to 255 characters long. You may use the LINEFEED key to continue the same program line on a new line on screen.

Bugs

1. The FILES output omits the first letter of the file extension if the file is Read/Only.
2. The exponentiation operator ^ calls the LOG function and therefore produces a value of only single-precision accuracy, even when used with double-precision variables.
3. The BASIC compiler's pseudo-random number generator is seriously flawed: FIX(RND*6) + FIX(RND*6) will never sum to 5, although it should do so one time in six. But the BASIC interpreter doesn't suffer from this bug.

POWER COMPUTING !

AT LOW, LOW PRICES !!

EXAMPLE SYSTEMS:

- | | |
|---|-------|
| 1. 286-12 MHz , 640 k , 1.2 Mb FLOPPY , 20 Mb HARD DISK | £1017 |
| 2. 286-20 MHz , 1 Mb , 1.2 Mb FLOPPY , 40 Mb HARD DISK | £1402 |
| 3. 386-20 MHz , 2 Mb , 1.2 Mb FLOPPY , 80 Mb HARD DISK | £2109 |
| 4. 386-25 MHz , 4 Mb , 1.2 Mb FLOPPY , 128 Mb HARD DISK | £2836 |
| 5. 386-25 MHz CACHE , 4 Mb , 1.2 Mb FLOPPY , 310 Mb HARD DISK | £4237 |

FOR

MONO SCREEN ADD
£130

EGA SCREEN ADD
£375

VGA SCREEN ADD
£628

"tower" CASE ADD
£95



ALL SYSTEMS INCLUDE

3.3 DOS AND GWBASIC

101 ATE KEYBOARD

1 YEAR ON-SITE MTCE

GM-6 MOUSE

WE'LL EVEN CONSIDER YOUR
OLD EQUIPMENT IN PART EXCHANGE

For specification sheets and further details, call

TEMPEST

TEL: (0272) 574 569

FAX: (0272) 701308

E.O.S. HOUSE
145 STAPLE HILL ROAD
FISHPONDS
BRISTOL BS16 5AG

ALL PRICES EXCLUDE V.A.T. , DELIVERY AND ARE CORRECT AT THE TIME OF GOING TO PRESS

Programming in BASIC for Beginners - Part 5

Additional Print Functions

Some applications require that certain expressions are not only printed on the screen but are printed in a particular place on the screen. Within XTAL BASIC there are a number of commands which allow the programmer to specify where on the screen his print expressions will appear.

PRINT@

This command allows the programmer to state exactly where on the screen his print expression will appear and works as follows. The screen is divided into 40 characters across and 23 lines down, so for the purpose of this command, the screen may be compared to the grids on a map. Like a map, by giving two coordinates we can specify an exact point on the screen. The syntax for the command is PRINT@ (horizontal position), (vertical position), EXPRESSION.

To illustrate this, key in the following :-

```
10 CLS
20 PRINT@0,0,"TATUNG EINSTEIN"
30 END
```

When RUN this will print the words TATUNG EINSTEIN in the top left hand corner of the screen.

Now key in :-

```
20 PRINT@12,0,"TATUNG EINSTEIN"
```

This will erase the original line 20 and replace it with the new one and when we re-RUN the program we will see that TATUNG EINSTEIN is printed on the top line of the screen 12 characters from the left.

Now key in :-

```
20 PRINT@12,10,"TATUNG EINSTEIN"
```

When RUN this time TATUNG EINSTEIN now appears at the 12th character along on the 11th line. I hear a voice in the back row say, "Why the 11th line when in our statement we specified 10 ?". This is because the first line on the screen is referred to as line 0. Experiment with changing the coordinates until you are familiar with the system.

PRINTTAB()

Those with a knowledge of typewriters will recognize the expression TAB. On the Einstein there are 40 TAB positions per line, TAB(0) being the extreme left position and TAB(39) being the extreme right. Using the TAB function we can therefore specify where on the line our print statement will appear. Note that more than one TAB command may be used on one line.

For example :-

```
20 PRINTTAB(5)"TATUNG"TAB(25)"EINSTEIN"
```

This will cause the word TATUNG to be printed at the 5th position on the line and EINSTEIN at the 25th position on the line. As an optional extra as it were, the first number inside of the brackets can be followed by a second number provided the two numbers are separated by a comma. This second number is the ASCII code number for a particular character. For example, the ASCII code for a full stop (.) is 46 so if we change our line 20 to read :-

```
20 PRINT"TATUNG"TAB(25,46)"EINSTEIN"
```

and RUN we should see a row of full stops between the words TATUNG and EINSTEIN. It should be noted that any following TAB function defaults to the specified character of the previous TAB function if this second figure is omitted. On power up this second figure is set to 32 which is the ASCII code for a space.

SEPARATORS

The spacing in PRINT expressions can also be controlled using a selection of SEPARATORS. Separators are symbols which can be positioned after a print expression. The semi colon (;) separating two expressions will cause the second expression to be printed immediately following the first, WITHOUT A SPACE.

Example :-

```
PRINT"TATUNG";"EINSTEIN" (will appear as TATUNGEINSTEIN)
```

If spaces are left within the inverted commas (") they will be considered to be character positions when the printout is executed.

Example :-

```
PRINT"TATUNG "; "EINSTEIN" (will now appear as TATUNG EINSTEIN)
```

THE COMMA

The comma (,) at the end of one expression indicates that the printing will restart at the next TAB POINT. Tab points are simply pre set positions on each line of the screen grid and are set 10 characters apart.

Example :-

```
PRINT"TATUNG", "EINSTEIN"
```

will cause TATUNG to be printed at position 0 and EINSTEIN to be printed at position 10.

It should be noted that after these simple commands have been mastered the user has quite a powerful facility for controlling the layout of the printed expression.

Dear Members

Well what a disappointing club night on the 11th of May !!!, only SIX members including myself turned up.

I seemed to detect a certain amount of apathy amongst the membership ?, it seems to be the same members who send in articles for the newsletter and turn up at club nights !.

If you are no longer interested in the club then please let us know so that the ones who are interested can stop banging their heads against a brick wall all the time, if you want to we can soon stop the club nights and turn the club into a bi-monthly newsletter club only ???, after all it's your choice !!!.

It would certainly make my job a lot easier but that would be a purely selfish attitude on my part !, on reading this newsletter you will have found out that I have a IBM compatible computer now but that doesn't stop me from still being interested in the Einstein (its still a great computer), so come on lets see if there is life after death because it appears that some of our members have passed away but forgot to tell anyone ?.

I know summer appears to be here at the moment and you are thinking of your holidays but try and think of something to say to me even if its only "go yourself", it would be nice to get some kind of response from some of you because if I don't I am thinking of packing it all in myself !!!.

If you wish to send me any articles, then send me them either on a disk or printed out on paper, at the moment I am trying to get use to using a desk top publishing program so that the layout of the newsletter will be more pleasing to the eye and I can add graphics to a page.

PLEASE DON'T IGNORE THIS PLEA AND LEAVE IT FOR SOMEONE ELSE

EVENTUALLY THERE WON'T BE SOME ONE ELSE .

***** SUPASOFT *****

CASH TRADER	Multi-choice, Multi-player Trading Game	£7.50
EAZIDRAW	Versatile Printed Circuit Board Designer	£12.50
SUPASHAPE	Shape/Sprite Design Suite with Library	£9.50
THE I CHING	Chinese Divination + Book	£9.50
SUPAPLAY	Nine Game Compendium	£7.50
SUPADISC	SUPAMATH + SUPAMECH + BANKMAN	£15.50
	SUPAMATH + SUPAMECH	£13.50
	SUPAMATH + BANKMAN	£11.50
	SUPAMECH + BANKMAN	£11.50
SUPAMATH	Maths tutorial and demonstration	£9.50
SUPAMECH	Mechanical Engineering tutorial	£9.50
BANKMAN	34k of financially oriented programs	£6.50

***** NEW ***** FREE *****

SUPACOMP Send a formatted 3" disk in a padded envelope, complete with return postage and receive this menu driven selection of 54 X BASIC programs free of charge. Topics covered include Mathematics, Statistics, Electrical Engineering, Physics and Surveying.

SUPACOMP is, and can be, freely distributed on the understanding that it is in no way altered and that it retains its copyright at all times.

Those finding the compilation of use, or interest, and who would like further information on the various topics contained therein can register their copy by sending a cheque, or postal order, made payable to SUPASOFT, to Chris Pickles at the address given below. They will be sent a copy of the SUPACOMP data book containing background information, including diagrams, pertaining to the various programs used in the compilation.

The main menu is divided into 5 sections:-

Mathematics / Electrical Engineering / Statistics / Physics,
Chemistry and Medicine / Navigation and Surveying.

With each section loading separately, as a single entity and containing at least 10 sub-topics.

All prices include postage and all orders and enquiries should be addressed to :- SUPASOFT, c/o Chris Pickles, 474 Hertford Road, Edmonton, London N9 8AD.

AVON EINSTEIN USER GROUP COMMITTEE MEMBERS

CHAIRMANMIKE IVORY, 1 HEATH ROAD, HANHAM, BRISTOL, BS15 3JT.
TELEPHONE No. 0272-616281

SECRETARYJOHN NASH, 38 BURFOOT GARDENS, STOCKWOOD, BRISTOL, BS14 8TE.
TELEPHONE No. 0272-838028

TREASURERROLF GEORGE, 25 FAIR VIEW, CHEPSTOW, GWENT, NP6 5BX.
TELEPHONE No. 02912-4916

LIBRARIANGRAHAM HIGGINS, 31 LENA STREET, EASTON, BRISTOL, BS5 6BD.
TELEPHONE No. 0272-559874

EDITORBOB SMITH, 2 INGLETON DRIVE, WORLE, WESTON-SUPER-MARE,
AVON, BS22 0SR.
TELEPHONE No. 0934-517465

Any articles, listings, reviews, questions, complaints, !! please send them to the Editor.

For all Public Domain enquiries !! please contact the Librarian.

PLEASE REMEMBER IF YOU ARE SENDING DISKS, AND YOU WANT THEM
RETURNED ? THEN ENCLOSE AT LEAST 28p RETURN POSTAGE

(Return address label if possible)

PLEASE NOTE.....All articles and programs become the copywrite of the
A.E.U.G. as well as the original authors !!!

Front cover graphics designed on First Publisher and printed on a
Olivetti DM280 dot matrix printer