VOL. 1
NO. 10

STATUS 1500

NOVEMBER
1983

* 12 issues £10.50 (UK) * overseas £14.50 * published monthly *

* Edited by RONALD COHEN, 62 BEENHEIM CRESCENT, LONDON W.11 *

THE MIXTURE AS BEFORE

"Keep up the good work!" is the theme of many of the letters I have received. I cannot avoid replying: "The same to you!" It is the ideas, queries, complaints, information and programs that YOU send in that keep this newsletter going. At the moment, feedback is falling off, and some of those whose letters and programs have been a source of inspiration are resting from the labour of stoking the insatiable fires of STATUS 1500. The attempt - not always successful - to maintain a very high standard of quality and relevance must inevitably mean that some material is rejected, or its use postponed indefinitely, until it ties in with what is being dealt with. Perhaps I owe more to these "mute, inglorious Miltons" of the keyboard, than I can adequately express. Material which is unprinted is as useful a contribution as that which is actually published. Selection is what editing is about. No one can edit a vacuum.

* * * * *

Naturally, as soon as I had written the paragraph above, I received several very interesting letters and contributions. In particular, I received some very interesting information about hardware from SIMON COX who has recently visited Japan: some of this information is summarised on page 101. He tells me that over there the Sharp PC 1500 is well supported and displayed, and is one of the most popular machines: there is an enormous quantity of software available. At least this confirms the suggestion that the PC 1500 is not being abandoned by SHARP, even if SHARP (UK) are less enthusiastic.

* * * * * *

DISAPPOINTMENTS - Sorry, no PEEK & POKE this month. This feature will resume in January. - Sorry, no complete memory map yet. I am working on this, but it takes time.

* * * * * *

ERRORS - In some copies the last line of the 'demo' program on page 92 ends *GOTO 20* . It should of course end *GOTO 4*
The Machine Code Chart on page 61 , column 9, row 3, reads BCI,i(-) in some copies. It should read BCS,i(-) .
Please amend your copy if necessary.

*** *CONTENTS* ***

94 THE MIXTURE AS BEFORE
95 SIGNALS
96 BUGS
97 RESTRUCTURE
98 Simple SUPERMERGE
98 "ANYBASE CONVERSION"
98 MINDBOGGLE CORNER
99 SUPERSKETCH
100 CAPSET reviewed
101 MARKET PLACE

IAN TRAYNOR is preparing a very interesting new program, and has had some difficulty with Error-Trapping for an EVAL routine. He finds that ON ERROR GOTO does not react to ERROR 1, and would like to know if this is standard.

Yes. ON ERROR GOTO acts when the internal search is unsuccessful. However, in the event of Syntax Error, the search cannot even start.

JOHN MACK writes plaintively that he finds the PEEK,POKE & MEMORY series 'terse almost to the point of ambiguity'. C.P.UNDERWOOD also finds it frequently unclear, but blames this on verbosity and repetitiveness.

I am glad you are both enjoying the series.

SYDNEY LENSSEN says that he obtained "SIXPACK" from DDD Software, on the recommendation of the review (page 70)in this newsletter, and he is very pleased with it.

Several other readers have made the same comment.

K.SOUTHGATE asks if one can CLOAD a program so that it will commence execution immediately it has been CLOADed. He points out that many computers have this facility.

I cannot see any way of doing this directly without alterations to the ROM. However the same effect may be obtained in the following manner:

a] Set up cassette-recorder in usual way, depress PLAY key.
b] Write NEW.
c] In PRO mode, write a 1-line program: 1:CHAIN
d] In RUN mode, key RUN [ENTER]

The program on your cassette will then CLOAD, and commence execution immediately it has CLOADed, without needing any further intervention.

CHRISTOPHER LEDSAM asks if he is right in thinking that assembly language is the same for most micros, and whether there is a learning book on assembler in general.

This topic was discussed on page 46, and again on page 60.INSTRUCTIONS DIFFER FROM CHIP TO CHIP, SINCE THEY HAVE DIFFERENT FACILITIES. SOME TERMS ARE THE SAME. OTHERS ARE DIFFERENT. With regard to your question about a book, you asked the same question in May, and I answered it in May, on page 32. Indeed, I wish there were such a book, but wishful thinking gets one nowhere, particularly in computing.

K. SOUTHGATE also asks me to withdraw his recommendation of the SMITH-CORONA typewriter-printer, which he was using with the CE 158 interface. He says that the printer has given a great deal of electrical and mechanical trouble.

JOHN WARNER has difficulty with programs from the PC 1500 "Applications Manual". He wonders whether the fault is in his computer, or in the programs.

Almost certainly in the programs. SHARP's programs are notoriously bugridden.

D.C.BOWRING is successfully controlling machine tools with the PC 1500 via CE 158.

We hope to print an article on this brilliant development at a later date. Meanwhile, for any reader who is seriously interested, Mr.BOWRING will kindly send copies of his notes: if a cassette is provided, relevant programs will be recorded onto it. (Dont forget SAE).

BUGS

It is reckoned, even among professional programmers, that it takes 5 times longer to debug a program than to write it. It certainly takes 5 times as much effort. It is very seldom that one can write a program that runs perfectly first time.

The causes of bugs are legion. Easy to cure are simple errors of syntax, and misprints, or omissions and errors of punctuation. These will usually show up as ERROR 1 etc., when you RUN the program. Slightly less easy to deal with are those which show ERROR in a line which is not actually where the error is caused. For instance, an error in a DATA statement does not show in the line which holds the DATA statement, but in the line which READs it.

Other bugs can be connected with FOR-NEXT loops. A statement could be inside a loop, whereas it should be outside, or vice versa. More commonly, it is at the beginning of the loop, instead of at the end (or vice versa) so that it repeats once too often, or too few times. Beginnings and endings, entrances and exits, can cause a lot of trouble.

Another source of bugs lies in the use of variables. If you INPUT a constant, and then multiply, divide it or whatever, it is useless to expect it to have the same value as when you originally INPUT it.You must do something like:

```
INPUT "N="; N: M=N
```

Then you can still recover M which will retain the original value of N.

Very much harder to cure are subtle errors of logic. DO NOT EXPECT THE COMPUTER TO READ YOUR MIND! (But - alas - one so often does!)
I had an example of this when trying to translate parts of "DELETE" into machine-code. It appeared to work with either of the 2 m/c routines, but not with both: although there appeared to be no way in which they should interfere with each other. The fault - which only occurred intermittently - was finally traced to the 'address-finding' routine. Having found the first part of a line number (in 256ary), the routine then looked for the 2nd part. Simple and logical? Not so! Remember, the 2nd part would be in the form of a number 0-255. Before it came to the 2nd part, it often found a similar number as part of the coding, and reported "Found!". I *wanted* the routine to find part of a line number, but I had only *instructed* it to find a specific number. I had expected it to read my mind. (The cure, of course, was simple: only to report when the routine had found 2nd part *immediately* following 1st part, otherwise to *restart* search).

Bugs that often go unnoticed, particularly in a complex program, are caused by the act of debugging. For instance, you might DIMension a certain number of variables - say DIM A$(65). Later you find you have insufficient memory, as a result of extending the program, so you change this to DIM A$(60). You have forgotten that an obscure and seldom-used subroutine still refers to A$(65). Result: ERROR. Whenever you alter anything to cure a bug, *check that you have not caused another one.*

Tracing bugs is not easy. Sometimes it must be done the hard way: LPRINT the value of every relevant variable after every single line. The error should show up. More often racking your brain helps not at all: you are still in the grip of an unjustified assumption (as I was with'DELETE'). Often the only answer is to put the program away for a time: the simple answer may come to you. Very often it is the simplest causes which are hardest to find. No one can do your thinking for you, least of all your computer, which can only obey. Make sure the instructions you give it lead to the *actions* you wish it to perform: not to the *results* you desire. If you expect it to produce results, you may well omit an essential step. Let me repeat "DO NOT EXPECT THE COMPUTER TO READ YOUR MIND!"

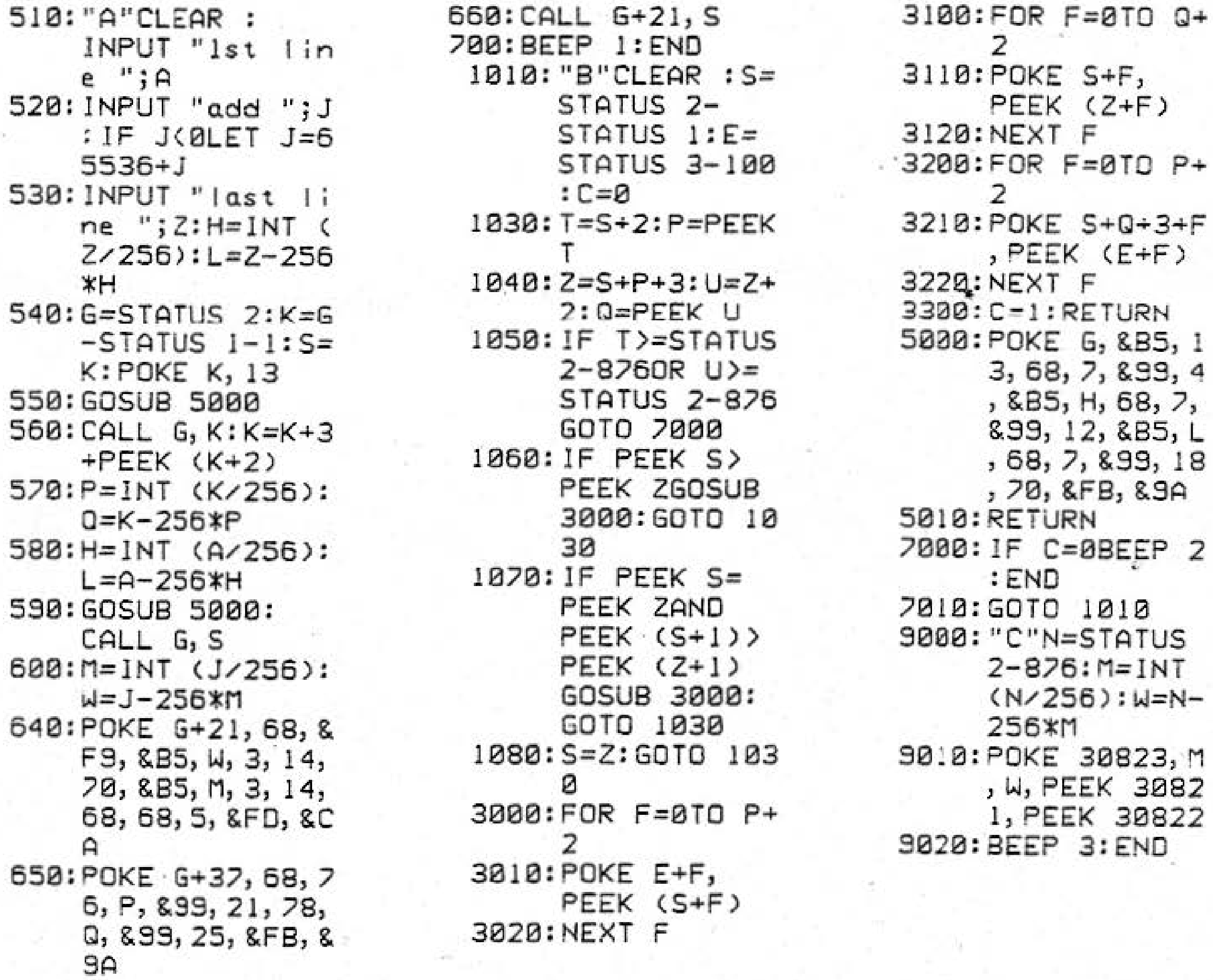
RESTRUCTURE

This program should be MERGEd with the subject program. You need 1K free.

DEF A allows you to Renumber a block of the subject program, without altering the intervals between lines, by adding a set amount to every line. e.g. lines 1,2,3,4,....99 could be changed to 501,502,503....599, by answering the INPUT prompts with 1,500,99. The converse is also possible, by answering the "ADD?" prompt with a minus figure. It MAY be advisable to limit all line numbers and additives to +/- 32767, although no difficulty has yet been encountered. Time to execute (any length): about 10 seconds.

DEF B will re-arrange the subject program in accordance with the newly-alloted line numbers. Thus, in the example above, lines 1,2,3 newly called 501,502,etc, will be placed after any lines with a lesser line number. WARNING: this procedure can be extremely slow. Time taken to transfer about 20 lines from the beginning to the end of an 8K program was 6 hours! You could of course leave it running overnight? (A machinecode version which should run faster is in course of preparation).

DEF C deletes the utility, and leaves the subject program on its own.



```
510:"A"CLEAR :
     INPUT "1st lin
     e ";A
520:INPUT "add ";J
     :IF J<0LET J=6
     5536+J
530:INPUT "last li
     ne ";Z:H=INT (
     Z/256):L=Z-256
     *H
540:G=STATUS 2:K=G
     -STATUS 1-1:S=
     K:POKE K,13
550:GOSUB 5000
560:CALL G,K:K=K+3
     +PEEK (K+2)
570:P=INT (K/256):
     Q=K-256*P
580:H=INT (A/256):
     L=A-256*H
590:GOSUB 5000:
     CALL G,S
600:M=INT (J/256):
     W=J-256*M
640:POKE G+21,68,&
     F9,&B5,W,3,14,
     70,&B5,M,3,14,
     68,68,5,&FD,&C
     A
650:POKE G+37,68,7
     6,P,&99,21,78,
     Q,&99,25,&FB,&
     9A
660:CALL G+21,S
700:BEEP 1:END
 1010:"B"CLEAR :S=
       STATUS 2-
       STATUS 1:E=
       STATUS 3-100
       :C=0
 1030:T=S+2:P=PEEK
       T
 1040:Z=S+P+3:U=Z+
       2:Q=PEEK U
 1050:IF T>=STATUS
       2-876OR U>=
       STATUS 2-876
       GOTO 7000
 1060:IF PEEK S>
       PEEK ZGOSUB
       3000:GOTO 10
       30
 1070:IF PEEK S=
       PEEK ZAND
       PEEK (S+1)>
       PEEK (Z+1)
       GOSUB 3000:
       GOTO 1030
 1080:S=Z:GOTO 103
       0
 3000:FOR F=0TO P+
       2
 3010:POKE E+F,
       PEEK (S+F)
 3020:NEXT F
3100:FOR F=0TO Q+
      2
3110:POKE S+F,
      PEEK (Z+F)
3120:NEXT F
3200:FOR F=0TO P+
      2
3210:POKE S+Q+3+F
      ,PEEK (E+F)
3220:NEXT F
3300:C=1:RETURN
5000:POKE G,&B5,1
      3,68,7,&99,4
      ,&B5,H,68,7,
      &99,12,&B5,L
      ,68,7,&99,18
      ,70,&FB,&9A
5010:RETURN
7000:IF C=0BEEP 2
      :END
7010:GOTO 1010
9000:"C"N=STATUS
      2-876:M=INT
      (N/256):W=N-
      256*M
9010:POKE 30823,M
      ,W,PEEK 3082
      1,PEEK 30822
9020:BEEP 3:END
```

FROM THE KEYBOARD

Did you realise that the instruction GRAPH will move the pen to the 'normal' position at the side, and reset the X and Y counters, even though you are already in GRAPH mode?

SIMPLE SUPERMERGE

After the EOL marker 13, at the end of every program, comes EOP marker 255 This intervenes between the first program and the MERGEd program, and prevents the pair running as a single unit. The answer, as several readers have kindly pointed out, is to load the MERGEd program one byte earlier, wiping out the intervening 255 . This may be acheived by subtracting 1 from the End-of-Program pointers. The End-of-Program address is held in 30823 and 30824. However, subtracting 1 from the contents of 30824, if 30824 holds 0, requires 'carry one' as in ordinary arithmetic . So in this case you would put 255 in 30824, and subtract 1 from 30823. This must be done *before* MERGEing the 2nd program. The following routine takes care of both variations, and resets EDIT/SEARCH pointers to beginning of 1st program:

FROM THE KEYBOARD   *[before MERGEing]*

```
a] A=PEEK 30823 [ENTER]
b] B=PEEK 30824 [ENTER]
c] POKE 30823, A-(B=0),(B-1)*SGN B + 255*(B=0),PEEK 30821, PEEK 30822 [ENTER]
```

ANY BASE TO ANY BASE CONVERSION

by George Cook

```
 10: CLEAR: INPUT"From base ";B1
 20: INPUT "To base ";B2
 30: INPUT "Enter No. ";N$
 40: L=LEN N$: ON ERROR GOTO 60: C=0: B$=CHR$ 0
 50: DIM N(L)
 60: FOR Z=L TO 1 STEP -1
 70: N(Z)=ASC MID$ (N$,Z,1)-48
 80: IF  N(Z)>9 LET N(Z)=N(Z)-7
 85: C=C+N(Z)*B1∧(L-Z)
 90: NEXT Z
100: X=-1: D=C
110: D=INT (D/B2):X=X+1
120: IF D>0 GOTO 110
130: FOR Z=0 TO X
140: D=C/B2: C=INT D: D=(D-C)*B2
170: D=48+D: IF D>57 LET D=D+7
180: A$=CHR$ D: B$=A$+B$
200: NEXT Z
210: ON ERROR GOTO 300
220: LPRINT N$;" = ";B$
230: INPUT "more?(Y/N)";Q$:IF Q$="Y" GOTO 30
240: END
300: PRINT  N$;" = ";B$: GOTO 230
```

This program will convert hex,decimal, binary, or any other base to any base. Printer or screen may be used. For a series of numbers with identical conversions the length of each number must be the same: expand if necessary by adding 0 at the beginning.

MINDBOGGLE CORNER

According to the Technical Reference Manual it is possible to alter the CSAVE and CLOAD speed. Write a program which will enable FAST CSAVE and FAST CLOAD.

***** £50 cash prize! *****

Naturally with so substantial a prize there are stringent conditions. Send for free entry form if you are seriously interested.

Closing date for competition: 7 January 1984

SUPERSKETCH

Always initiate execution by RUN. To draw a picture or design answer the prompt DRAW/COPY by D. Use the numeric keys 1-9 as cursor controls to move the pen across the paper.

*To change the type of line, or the color,press key 5. Normal lines are type 0. Various degrees of dottedness are types 1-8. Type 9 is blank, for moving the pen to another part of the paper. In this case, you are also asked the size of each step. To alter this, you must press 5 again; you must also key 5 again to continue the picture. For all lines other than blank type 9 the size of the step is 1+2*T [line type] You may answer prompt type with T if you do not wish to change; C for same color. The size of the design is limited to + or - 255 on the Y axis, from the point of the commencement. The screen shows X position, Y position, line type, and color. ALWAYS END EVERY DESIGN BY PRESSING THE / KEY. THIS IS ESSENTIAL, If interrupted during a change, resume by GOTO 53. If you wish to continue an interrupted drawing, key GOTO 61.*

To inspect a drawing during execution, key [.] Restore by [=] key. To repeat a drawing, answer prompt DRAW/COPY with C. As you draw a picture the program actually amends itself to draw this picture. If after drawing a picture, you CSAVE the program, you will notice it is much longer than the original. The program has been extended to include all pen movements, which are recorded in the"nomansland" following STATUS 2. RUNning the recorded program restores the original STATUS 2, without affecting the information which can still be read on COPYing.

To CSAVE the amended program set up the cassette-recorder in the usual way, then key DEF K. This will offer a prompt for the file-name, and extend the end-of-program counters, and then CSAVE.

```
  1:GRAPH :GOSUB 8
    00:Z=S+1
  2:INPUT "draw(D)
    /copy(C)";C$:
    GOTO C$
  5:"D"CLEAR :S=
    STATUS 2:Z=S+1
  7:"A"ON ERROR
    GOTO 9:WAIT 0:
    D=0:IF T=9
    INPUT "distanc
    e ";D
  9:PRINT X;Y;T;C:
    A$=INKEY$ :
    GOTO ASC A$
 46:GLCURSOR (X,Y-
    150):GOTO 9
 47:POKE Z,9,9,9,9
    ,9:GOTO 90
 49:GOSUB 129:X=X-
    J:Y=Y-M:GOSUB
    555:GOTO 9
 50:GOSUB 129:Y=Y-
    M:GOSUB 555:
    GOTO 9
 51:GOSUB 129:X=X+
    H:Y=Y-M:GOSUB
    555:GOTO 9
 52:GOSUB 129:X=X-
    J:GOSUB 555:
    GOTO 9
 53:WAIT :CLS :
    INPUT "type ";
    T:INPUT "color
     ";C:T=T*(T<10
    ):C=C*(C<4):
    GOTO 7
```

```
 54:GOSUB 129:X=X+
    H:GOSUB 555:
    GOTO 9
 55:GOSUB 129:X=X-
    J:Y=Y+K:GOSUB
    555:GOTO 9
 56:GOSUB 129:Y=Y+
    K:GOSUB 555:
    GOTO 9
 57:GOSUB 129:X=X+
    H:Y=Y+K:GOSUB
    555:GOTO 9
 61:"="GLCURSOR (X
    ,Y):GOTO 9
 90:"Z"BEEP 3:END
129:Q=1+2*T*(T<9)+
    D
130:H=(X<210)*Q:J=
    (X>0)*Q:K=(Y<2
    55)*Q:M=(Y>-25
    5)*Q:RETURN
525:X=PEEK Z:Y=(
    PEEK (Z+1)-128
    ))*PEEK (Z+2):
    T=PEEK (Z+3):C
    =PEEK (Z+4)
555:"L"LINE -(X,Y)
    ,T,C
560:IF R=0POKE Z,X
    ,128+SGN Y,ABS
    Y,T,C
570:Z=Z+5
575:RETURN
```

```
700:"C"GRAPH :ON
    ERROR GOTO 90:
    R=1:Z=S+1:
    GOSUB 800
705:X=PEEK Z:Y=(
    PEEK (Z+1)-128
    )*PEEK (Z+2):T
    =PEEK (Z+3):C=
    PEEK (Z+4)
710:GOSUB 555:GOTO
    705
800:S=256*PEEK 308
    21+PEEK 30822+
    0983
805:POKE S-2,13,25
    5
810:POKE 30823,(S-
    1)/256,S-1-256
    *INT ((S-1)/25
    6)
815:RETURN
900:"K"Z=Z+5:POKE
    Z,13,255:S=Z+2
    :POKE 30823,
    INT (S/256),S-
    256*INT (S/256
    )
905:INPUT "title "
    ;N$:CSAVE N$
910:REM 1234567890
```

This is a most ingenious program. It will produce display lettering, under user control, in a variety of shapes and sizes. Some of the effects that can be created may be seen in the advertisement on page 93 . Error-trapping is verbal, and the required sequences of action are adequately prompted.Additionally, automatic justification is available.

With so many virtues in the program, it may seem surprising that I find it hard to commend it. In my opinion it is almost spoilt by a plethora of petty faults. If indeed these criticisms seem petty and carping, I am obliged to answer that meticulous attention to petty detail is the essence of good clear typography, particularly for display purposes. The viewer is not normally required to admire the typographer's skill: merely the impact of his efforts. Here the programmer's efforts are considerable; the impact less so.

Being limited to 10K,only one style of lettering is available, something like "*COMPACTA BOLD OUTLINE*" .'Outline' and 'Shadow' faces are among the hardest to handle effectively, since the weight of each letter is visually thrown outwards. The spacing between letters is always of the most critical importance in typographic design, here particularly so. And it is over this one factor that the user has no control at all! Indeed, even the automatic justification, when used, does not control the spacing of the letters, but the width of the letters themselves! The spacing is adequate in the larger sizes, but in the smaller sizes the letters run into each other, so that the only weight is between the letters, destroying any effective visual impact.

Of the total available memory, 1½K is occupied by variables, 5K by data for the design of letters, and 3½K for programmable manipulation, leaving only 150bytes free; thus there would be no memory available for control of spacing, important as it is. The answer of course is to sacrifice some of the more elaborate facilities, in order to create space. For instance, the very clever error trapping, which will translate ERROR 80 and ERROR 78 into a form of words saying "LOW BATTERY" is not really essential.

I also found difficulty in dimensioning the letters. This was exacerbated by the fact that measurements are in 'points', which are not the same as the'point sizes'normally used in printing, but refer to pen movements of the CE150. I did not find it easy to cope with the cursor movements, which were 9 times as great in one direction as another. I did not find it easy to collate the placing of lettering, which is initially visual, with the size of lettering, which is by calculation. The author admits that the use of the cursors, when changing from horizontal to vertical layout, may be confusing. In fact the only confusion is caused by the explanation itself. It would have sufficed to say that the cursor-movements are orientated with the lettering, not with the paper-movement.

Nor can I say I was entirely happy with the combination of menu-drive and command-drive. Where there is - as here - a screen prompt offering a number of choices, one has become accustomed to pressing the function-key beneath the appropriate choice. In this program (and I think an unnecessary complication) one must press a key corresponding with the initial letter of one's choice. And the result of pressing a wrong key is the brief but tactless error message "WHAT".

Nevertheless the program has potential. I hope that the author may revise it, and produce a second version with perhaps less ingenuity and more use. At the moment its price though not unreasonable,reflects the great effort that has gone into creating the program, rather than its value to most users.

CAPSET, by Charlie Simpson, is available from DDD Software, at £9.50

MARKET INFORMATION

The population of Japan appears to be divided into 2 very distinct groups: a) Those who understand computers.
b) Those who understand English.
In consequence, obtaining full and precise information is not easy. However SIMON COX, on a brief visit to Japan, reports some very interesting developments.

The PC 1501 (also known as PC 1500A) seems to be identical with our PC 1500, except that the built-in memory has been enhanced by 8K. It should be compatible with all the present add-ons.

Even the PC 1502 has now recently appeared! This, again,should be compatible with PC 1500, but *may* have enhanced ROM. Via the CE 158, it can be connected to a 200K microdiscdrive (price about £200), and with the addition of a unit GP40 (price about £40) can be connected to a VDU or television. It is not yet certain whether UK televisions are compatible, and it is believed that one needs the CE 158 before the GP 40 can be connected.

Anyway, the PC 1500 system is booming in Japan, so it should not become obsolete for some years yet!

* * * * * * * * * * * * *

The new catalogue from WALTER SPIEDEL (address on page 71) contains some interesting items.Unfortunately the catalogue is in German, and a query to SPIEDEL as to the possibility of his sending out anything in English has not met with .. a reply.

TOOLKIT 2, which gives fast CSAVE and CLOAD, costs 371 DM. It is about the size of a packet of cigarettes, and plugs either into the CE 150, or direct into the PC 1500 in place of the printer. (But how would one use it without CE 150?). 8K EPROMs , and an EPROM programmer, are also featured. So is the "EXPERIMENTER" (498 DM), the"HEXMONITOR" (98 DM), and the "MACROASSEMBLER" (about 178 DM). For the latter one must specify what type of memory extension one has. The program is 4.5K long.

* * * * * * * * * * * * *

From ATLANTIC NORTHEAST MARKETING (address on page 71 also) comes an equally interesting catalogue. Featured is the mysterious CE 160, which appears to be a programmable EPROM. An EPROM burning service is offered: prices range from $99 to $78 according to quantity. Whether this service includes the module itself, or is extra to the purchase ofit, is not quite clear. Also featured are the SHARP ROMware Libraries. These are plug-in modules, each containing *up to* 16K of programs on specific subjects, such as statistics, maths, engineering, circuit analysis, and finance. The series is CE501A to CE507A.

* * * * * * * * * * * * *

I have not yet seen TANDY's book *"GETTING STARTED WITH YOUR PC2"*,but I am told that it is extremely helpful, particularly for someone who has worked through the Instruction Manual, and wants to know *"where do I go from here?"* I hope to review this in December or January.Obtainable from TANDY COMPUTER CENTRES at £8.99. TANDY at Centrepoint have the book in stock (address on page 43).

* * * * * * * * * * * * *

ELKAN ELECTRONICS inform me that mailing of the new edition of their "Pocket Computer File" has been delayed, but it should be sent out during the next few weeks. If you are not on their mailing list already, write to the address given in their advertisements in this newsletter.

* * * * * * * * * * * * *

I am told by IAN TRAYNOR that he is writing a most interesting program on *LINEAR PROGRAMMING* , which can save its cost many times over to some business users. It will be marketed by ELKAN, and we hope to review it in STATUS 1500.