VOL. 2
NO. 1

STATUS 1500

FEBRUARY
1984

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

Edited by RONALD COHEN, 62 BLENHEIM CRESCENT, LONDON W.11

VOLUME 2

Welcome again to a new volume of STATUS 1500. Mostly, it will be just 'the mixture as before'. The new series, 'Lets write a program' will be of interest to beginners and, I hope, to those more experienced who will welcome the opportunity to show that they know better than I do about programming techniques. A new series of subroutines, both in Basic and machine-code, depends on *YOU!* Some busy experienced programmers have promised programs which have never materialised. Some readers have the impression that, having paid their subscription, they can now sit back and enjoy the magazine. Others, expert programmers, will not supply material without payment. They consider they are entitled to reap where they have not sown. So perhaps I should emphasise that your subscriptions cover the costs of production and postage, and most expenses. If you want useful articles, good programs and so on, it is up to *YOU* collectively to supply the material for me to edit - I admit, sometimes ruthlessly! Of course, many readers are not able to supply this material: but even these quite often have queries and problems: this is all grist to the mill, and keeps the newsletter going.

At this point I would like to express my thanks to those who have written recently with programs and information, most of which I hope to print, and all of which is useful. Many of you have expressed their satisfaction with the high standard usually maintained by the newsletter. This high standard does depend on a certain greed for the obtaining of material, and severity in dealing with once it is received. So let me say again, that even if your material is not used immediately, or even not at all, it is always helpful, and always appreciated.

New readers will soon get used to the mixture of fulsome flattery and dyspeptic bad temper which appears to permeate this column. Some readers, who have been with the newsletter since early days, find it unpalatable when I 'get on my high horse' when something displeases me. But one of the reasons I started this newsletter was so as to be able to fulminate to a captive audience. (Captive till next January, anyway). New readers may find it useful to have bound volumes of volume 1. These will be available about the end of February, price £8.50 in UK, or £13.50 overseas, postage included. Happy programming!

SIGNALS

J.A.B.MAYNE has difficulty running the GOLF program from the December issue. The program stops at line 5940. He believed originally that there was a fault in the program; but after reading reports in the January issue now assumes he has made a typing error. He wonders if other readers have found a BREAK in the same place.

It is really disheartening. Again and again I have made it clear that all programs are rigorously tested before publication; if a program will not run it is almost inevitably a typing error on your part. Again and again I have pointed out that I cannot deal with such mistakes unless you enclose a listing of what you are trying to run, and it seems to me totally unreasonable - to put it mildly - to expect me to consider such problems without bothering to give me the necessary information.

R.J.COURT has difficulty running the GOLF program from the December issue. All goes well until he reaches the green, but the ball will not then enter the hole.

There are various possible causes - plus angles instead of minus, or using the wrong club such as 8 or 9, for woods and bunkers (see rules) which distort the effect of the stroke. Or perhaps there is a typing error. I CANNOT HELP YOU IF YOU DO NOT ENCLOSE A LISTING.

IAN TRAYNOR reports that his cat was overcome by the Xmas festivities, and treated the PC 1500 with disdain.He believes that readers may be happy to know that the keyboard of the PC 1500 appears to be waterproof.

SIMON COX regrets the delay in producing his booklet on Memory Extension. A disaster to his original manuscript, caused by the carelessness of the printers, has obliged him to rewrite the entire manuscript. However he confidently hopes that the booklet will be ready and available within a month.

ELKAN ELECTRONICS announce that they have relinquished the agency for the products of WALTER SPIEDEL, owing to difficulties in communication.

ALLAN THOMAS has started a PC 1500/PC 2 "USER GROUP" in New Zealand. It only has a few members so far, but is growing rapidly.

Readers in that part of the world who are interested should write to: ALLAN THOMAS, P.O.BOX 155, NAPIER, NEW ZEALAND.

H.TANG and R.CHAN have discovered that if you SJP to &E42C you will get the INKEY$ function. The ASCII code of the character keyed will be stored in the Accumulator. The contents of resisters XH, XL, and UL will be altered. They hope that this information may be of use. They add that they would like to see more m/c programs in the newsletter.

So would I! but I must rely on readers whose experience of machine code has been gained on other machines to supply such programs in a form which is useful and comprehensible to those less experienced with machine-code. The information you give sounds interesting; but can it be made use of in such a way as to avoid the sluggishness which usually accompanies the use of the INKEY$ function?

THE EDITOR BEGS CONTRIBUTORS TO WRITE THEIR NAME, AND THE SUBJECT, ON ALL CASSETTES, AND ON ALL LISTINGS. IF A LISTING IS IN SEVERAL PARTS, THE SAME APPLIES TO EACH PART. SO MUCH EXCELLENT MATERIAL IS SADLY WASTED BECAUSE IT CANNOT BE IDENTIFIED.

MORE SIGNALS

PETE ELDRIDGE has written a very interesting program for indexing and labelling program cassettes - unfortunately too long to print here. Still working on 'Password'protection, he wonders whether it would be possible to disable the RESET button.

This is hardwired, I believe, and should not be touched.

J.A.B.MAYNE has now keyed in the GOLF program again from the start, and has no problems.

Forgive the asperity of my earlier reply. I am glad the program is now working. Although such faults are almost always the result of typing errors by the reader, it is not totally impossible that a fault in the original program has been overlooked. Without a listing there is no way to check.

CLARENCE JOHNSON writes from Canada to say that he has found an American CALC-type program useful for many purposes. He adds that STATUS-1500 has helped him to keep pace with his highly-computerised family.

It is not always easy to bridge the gap between the younger generation who have been brought up with computers, and those who, like myself, have met these devices later in life.

FABRIZIO FESANI says that colleagues in Italy have got surprising results by substituting a 4KH quartz crystal for the original. CSAVEing for instance is twice as fast: but programs saved with original crystal cannot now be CLOADed. The clock however is not affected.

Has anyone else been working on this?

K.SOUTHGATE has now acquired a BBC computer. He wonders if any other reader has succeeded in interfacing the PC 1500 with this machine.

H-H.HEINE has difficulty in making his TANDY GP115 printer work with his PC 1500.

Write to: TANDY CORPORATION // "Customer Services" // Bilston Road // WEDNESBURY // WEST MIDLANDS // WS10 7JN

E.MACMILLAN informs me that he learns from USA that when you MERGE 2 programs, you cannot edit the 1st one, and that you must use DEF [label] to execute them.

This was explained in this newsletter last April [vol.i., p.23]

R.J.COURT, in New Zealand, has been impressed by the BROTHER EP 42 typewriter/printer, interfaced with PC 1500 via CE 158.

This combination is superbly portable. On the other hand, the print quality of the EP 42 I find barely tolerable.

Also in New Zealand, ALLAN THOMAS has been appointed New Zealand agent for the EASI- series of software.

Readers in that part of the world may find it more convenient to obtain this software directly from Mr. Thomas, rather than from UK. Address on previous page.

PETER NICOLS points out a further use of [SHIFT]/[CLEAR]. He says that this will clear a USING format, in the same way that RUN does.

Surely CLEAR does this also? But [SHIFT]/[CLEAR] is I suppose marginally faster, being 2 keys, instead of C L . and [ENTER].

YET MORE SIGNALS

J.K.GAUTON has difficulty using the RESERVE program (vol.i,p.7) with his CE 161. He asks whether the RESERVE has become positioned differently.

No. The RESERVE is still in the same place, relative to the memory. It is the Start-of-Program position which has moved, as a result of NEW 256. PEEK & POKE, page 5 , this issue, explains in greater detail.

TIM LANDON points out that the figures given on page 85, vol.i, for the start and end of the Basic stack, are not correct.

You are right. In fact the stack begins at &7A38 and ends at &7AFF

ARTHUR COX regrets that one cannot use quotation marks in SHARP Basic

Try this: *10: LPRINT "It is incorrect to say"; CHR$ 34;"Quotes are impossible!";CHR$ 34*

JAMES LOTHIAN finds incorrect my reply to his Signal in last December issue (vol.i,p.103). He says that it is not necessary to POKE values into 30823 to 30826, since these are taken care of by the NEW statement. With regard to the ROM information, at the 1st 7 bytes of memory, he believes that this is used only by battery-expanded modules, to mark the start of the undeletable area. There may be other purposes not yet discovered.

DAVID RIHOY has sent me a very interesting utility, for removing unwanted REM statements from a program. It will be printed in a month or two.

I have not yet used it on any REMs, but have found it most effective for getting egg off the ceiling.

JOHN MACK asks if it is possible to use PEEK & POKE techniques to discover the pen positions, and to control them.

Below are the relevant details from the Memory Map. It is certainly possible to PEEK these: and to POKE into the 'USER COUNTERS'. Indeed, the technique of unlimited Reverse Linefeed is described on page 24,vol.i. But it would be most unwise to POKE into the 'X-direction Absolute' and 'X-direction Scissoring' Counters:if you did so, this might try and force the pen to move laterally beyond its mechanical limit,and cause a mechanical failure. [XL and XH, etc, are in 256ary]

31200: USER COUNTER XH
31201: USER COUNTER XL
31202: USER COUNTER YH
31203: USER COUNTER YL
31204: SCISSORING COUNTER YH
31205: SCISSORING COUNTER YL
31206: ABSOLUTE POSITION X
31207: SCISSORING COUNTER XL
31208: SCISSORING COUNTER XH

The 'ABSOLUTE POSITION X' gives the actual position of the pen, 0 to 512, laterally. The 'SCISSORING COUNTER XL and XH' are added to this, to give the theoretical position of the pen, when this theoretical position is off the paper.

G.A.LATHAM points out that the note on 'CALL' (vol.i, p.79) does not mention that any variable used must be between the limits 32768 to -32769. He says that you can get round this by the following statement, which uses the sign bit to give the correct value in the m/c program:

10: IF A>32768 LET A=A-65536

Also from G.A.LATHAM comes a answer to the problem propounded by L.E.SIMONS in December Signals (vol.i., p.103), but his solution, which appears logical, must be held over until next month.

PEEK, POKE & MEMORY - XI

The idea mentioned in last month's PEEK & POKE, of keeping a record of all the System Pointers (when carrying more than 1 program in memory) by POKEing their contents into spare space in the Reserve Template, has aroused interest, so here is some amplification of the method.

It is useful to know, incidentally, that when you take a CE 161 or CE 159 out of the computer, and - on replacing it - key NEW 256, you do not destroy the contents of the Reserve or the Reserve Template. At the moment, with so many renewals taking place, I keep my Mailing List permanently in the CE 161, and use remaining space for other shorter programs. I have set up Reserve Keys on line 111 to read

POKE 13, PEEK 30823, PEEK 30824
POKE 30823, PEEK 13, PEEK 14
POKE STATUS 2-STATUS 1, 0
PEEK 30824=PEEK 24

The first line stores the values of the pointers indicating the end of the 1st program in locations 13 and 14. (The start-of-program and search/edit pointers are reset by NEW 256)
The 2nd line is used to restore these values to the end-of-program pointers on replacing the module, or reverting to the 1st program from a later one.
The 3rd line is used to replace 255 with 0 in the 1st byte of the first line number, since this had become 255 when I wrote NEW or NEW 256.
The 4th line is an extra check, before taking the module out. If keying this key,and ENTER, displays 1 , then I know that the EOP pointers are safely stored in the Reserve Template.
A similar system to the above may be used for succeeding programs: but for these all 6 system pointers should be stored - unless you are content to retrieve each one successively by setting up the first program, writing NEW STATUS 2, setting up the 2nd program, and so on.

* * * * * * * * * *

Some readers are confused about the exact location of the Reserve area, and its components, and how to reach back to it from the Program Area.

0 to 6 holds ROM information
7 to 85 holds the Reserve Template Area
86 to 196 holds the Reserve itself.

These figures are for the CE 161, whose memory starts at 0. For other modules whose memory starts later, add the relevant number of K. (1K=1024). For instance the memory of the standard machine starts at 16K, the CE 155 at 14K, and the CE 159 at 8K. So normally your Program Area starts 197 bytes after the relevant quantity of K.
However the CE 161 is something of a special case, since its instructions suggest that you should start your Program Area at 256 (by NEW 256) instead of at 197 (by NEW 0). One is advised not to use the region between 197 and 255 (heaven knows why!). So since the start of your Program Area is that much further on, you must reach back that much further. The start of your Program Area is, of course, with a program in memory, STATUS 2-STATUS 1.

It is not hard to check on the exact location of your Reserve proper, after you have worked out roughly where it is. Go into RESERVE mode, and write NEW. Then prime the lefthand key of line 1 as AB. If you PEEK around where you expect the Reserve to start, you will find that the 1st 3 bytes of Reserve are 1,65,66.

In fact, I am not quite sure about the figures above, 85 and 86 , for end of Template Area and beginning of Reserve; they might perhaps be 86 and 87 respectively. I do not have the computer with me as I write this, and shall not remember to check these locations. I'll let you do that.

LETS WRITE A PROGRAM! - I

It is not hard to start writing a program. It is not so easy to produce a streamlined foolproof finished article. Sometimes the difficulties in the later stages are caused by an intitial approach which was not adequate to the task. This series will examine some of these difficulties, and suggest some alternative strategies. The purpose is not intended to supply all the answers; but by bringing a few of the problems out into the open, it may be easier for you to find your own answers.

So let us pick a subject. During the coming months we can work on it together, and I hope that your criticisms and suggestions will be by no means the least valuable part of the exercise.

I suggest as a subject: ROULETTE. Now the very first question must be: *"Is this a suitable subject for a computer program?"* The answer, fortunately, is *"Yes!"* We will be dealing with such things as odds, probabilities, totals, random numbers, plus and minus totals, possibly some simple graphics; all eminently suitable for computers. The second question: *"Is 8K enough?"* Very probably. Roulette is not that desperately complicated. Choosing one of 36 random numbers is obviously simpler and quicker than asking the machine to examine all the possibilities in a game of chess, for instance. In fact, it could be so simple that 2K might be sufficient, thus making the game available to the unexpanded. Let us at the moment keep an open mind about this. A possible answer might be to keep the program within 2K, except for the graphics as an optional extra - a luxury version. At this point I should mention that I have not yet written a program on this subject: it is genuinely a "workshop project".

By now indeed you may be heard to mutter " *Lets stop talking and get on with it!*" NO! On the contrary, there is scope in this exercise for quite a lot more discussion before we write a line of code. It will pay off in the long run. Really it will. The more you know about what you are going to do before you start doing it, the more smoothly everything will run.(At least in theory). One of the difficulties often is that this forethought is impossible. Important ideas are stimulated by the work one does on the program: if one is inventing a game the rules follow the way it all works out. Often, particularly if one is a beginner, one *must* write a few lines of code to see if one's ideas are practical or not. This is sneered at by experts. Their views are not always relevant: and sometimes actually inimical to amateur programming.

Flowcharts are often recommended by books and by teachers of computing. In my opinion they are usually a nuisance, and obscure the real problems. They also tend to encourage a large quantity of sometimes unnecessary IF statements. They are fine as a notation for decisions already taken, but not much use as an aid to decision-making.They are also useful as an *analysis* of a program: they can highlight sequences which do not join up. One of the purposes of flowcharts is in connection with enormous projects where a whole team of programmers is working on different parts of the same program. This hardly concerns us. I do not intend to use a flowchart on this project of ours: if it becomes useful to do so, I can always change my mind. In any case, fashionable experts now favour what is called *"TOP-DOWN"* programming. All this means, very roughly, is sorting out what you intend to do before you start doing it.

A number of techniques have been developed for this pupose. The *WARNIER-ORR* system is one of them. I do not believe such elaboration is necessary , particularly where we are not working as independent members of a team, each doing a different job on a sort of assembly-line. On the other hand some sort of rough notation may be necessary. We can each develop our own, and maybe some of our ideas may be useful to each other.

GRAPH

by F.C.ODDS

Almost any equation, up to 64 characters in length, may be INPUT in answer to the prompt Y= _____. Up to 3 curves,of different equations, may be drawn on the same set of axes, for comparison.
IMPORTANT: Use as variable in your equation Z instead of the more conventional X.
Execute by DEF X. Points to watch: if you are drawing more than one curve on the same set of axes, use first the equation which has the biggest range of results. Note that if the smallest value of Z is not less than 1, the Y axis may not be drawn, but this will not affect the curve. Note also that if the least value of Z is 0, some equations may be imposible to resolve (giving ERROR 39).The answer is take a minimal value such as 0.00000001 for the smallest value of Z.

```
 10:Y(1)=XXXXXXXXX
    XXXXXXXXXXXXXX
    XXXXXXXXXXXXXX
    XXXXXXXXXXXXXX
    XXXXXXXXXXXXXX
    XX
 20:RETURN
100:"X"CLEAR :DIM
    A$(1)*50,B$(50
    ),B(54),X(50),
    Y(50):RR=0:MX=
    STATUS 2-
    STATUS 1+7
110:WAIT 150:PRINT
    "GRAPH DRAWING
     ROUTINE":
    PRINT "USE Z A
    S VARIABLE"
120:PRINT "Input d
   esired equatio
   n..."
130:INPUT "Y=";A$(
    0):L=LEN A$(0)
    :A$(1)=A$(0)
140:S=0:IF LEFT$ (
    A$(0),1)="J"
    LET S=2
150:FOR J=1TO L
160:B$(I)=LEFT$ (A
    $(0),I):B(I)=
    ASC B$(I)
170:A$(0)=MID$ (A$
    (0),2,L-1)
180:NEXT I
190:I=1:J=1
200:IF (B(I)>64)*(
    B(I)<89)GOTO 2
    50
210:POKE MX+J,B(I)
220:I=I+1:J=J+1:IF
    I>LGOTO 390
230:GOTO 200
250:P=B(I):Q=B(I+1
    ):R=B(I+2)
260:IF P=65IF Q=66
    IF R=83LET B=1
    12:GOTO 290
261:IF P=65IF Q=67
    IF R=83LET B=1
    15:GOTO 290
262:IF P=65IF Q=83
    IF R=78LET B=1
    15:GOTO 290
263:IF P=65IF Q=84
    IF R=78LET B=1
    17:GOTO 290
264:IF P=67IF Q=79
    IF R=83LET B=1
    26:GOTO 290
265:IF P=69IF Q=88
    IF R=88LET B=1
    28:GOTO 290
266:IF P=73IF Q=78
    IF R=84LET B=1
    13:GOTO 290
267:IF P=76IF Q=79
    IF R=71LET B=1
    19:GOTO 290
268:IF P=83IF Q=73
    IF R=78LET B=1
    25:GOTO 290
269:IF P=83IF Q=81
    IF R=82LET B=1
    07:GOTO 290
270:IF P=84IF Q=65
    IF R=78LET B=1
    27:GOTO 290
280:GOTO 320
290:POKE MX+J,24),
    B
300:I=I+3:J=J+2:IF
    I>LGOTO 390
310:GOTO 200
320:IF P=76IF Q=78
    LET B=118.GOTO
    340
321:IF P=80IF Q=73
    LET B=93:GOTO
    340
330:BEEP 3:PRINT "
    ERROR IN ARGUM
    ENT":GOTO 120
340:POKE MX+J,24),
    B
350:I=I+2:J=J+2:IF
    I>LGOTO 390
360:GOTO 200
390:POKE MX+J,58,2
    4),17)
400:IF RRGOTO 430
410:PRINT "Input v
    ariable range.
    "
420:INPUT "From: "
    ;R1:INPUT "to:
    ";R2:RX=ABS (
    R2-R1)
430:INPUT "colour
    (1, 2 or 3)? "
    ;CO:CLS
435:IF RRLET Q3=R3
    :Q4=R4
440:R3=0:R4=0
450:FOR I=0TO 50
455:BEEP 1,5,5
460:X(I)=R1+(I*RX)
    /50:Z=X(I)
470:GOSUB 10
480:IF Y(I)>R3LET
    R3=Y(I)
490:IF Y(I)<R4LET
    R4=Y(I)
500:NEXT I.IF RR
    GOTO 515
505:IF CS=2)*(ABS
    R4<ABS R3)LET
    R4=-R3
510:GRAPH :
    GLCURSOR (10,-
    300):SOREN :
    COLOR 0
515:IF RRLET R3=Q3
    :R4=Q4
520:RY=ABS (R3-R4)
    :P1=(200*R1)/R
    X:P2=(200*R2)/
    RX:P3=(200*R3)
    /RY:P4=(200*R4
    )/RY
525:IF RRGOTO 650
530:GLCURSOR ((200
    *(0-R1))/RX,0)
    :SOREN
540:LINE (P1,0)-(P
    2,0):LINE (P2,
    0)-(P2,-10):
    LINE (P1,0)-(P
    1,-10)
550:IF (R3>0)*(R4<
    0)LINE (0,P3)-
    (0,P4)
560:IF (R3>0)*(R4=
    0)LINE (0,P3)-
    (0,-20)
570:IF (R4<0)*(R3=
    0)LINE (0,P4)-
    (0,20)
580:LINE (0,P3)-(-
    10,P3):LINE (0
    ,P4)-(-10,P4)
590:CSIZE 1:
    GLCURSOR (P1-5
    ,-20):LPRINT R
    1:GLCURSOR (P2
    -15,-20):
    LPRINT R2
600:IF R1IF R4=0
    GLCURSOR (P1-1
    0,P3+10):
    LPRINT R3:GOTO
    650
605:IF R4=0
    GLCURSOR (-10,
    P3+10):LPRINT
    R3:GOTO 650
610:IF R1IF R3=0
    GLCURSOR (P1-1
    0,P4-10):
    LPRINT R4:GOTO
    650
615:IF R3=0
    GLCURSOR (-10,
    P4-10):LPRINT
    R4:GOTO 650
620:IF R1GLCURSOR
    (P1-10,P3+10):
    LPRINT R3:
    GLCURSOR (-10,
    P4-10):LPRINT
    R4:GOTO 650
625:GLCURSOR (-10,
    P3+10):LPRINT
    R3:GLCURSOR (-
    10,P4-10):
    LPRINT R4
650:COLOR CO:PX=20
    0/RX:PY=200/RY
660:GLCURSOR (PX*X
    (0),PY*Y(0))
670:FOR I=1TO 50
680:LINE -(PX*X(I)
    ,PY*Y(I))
690:NEXT I
700:IF S<>2GOTO 75
    0
710:GLCURSOR (PX*(
    X(0)),PY*(-Y(0
    )))
720:FOR I=1TO 50
730:LINE -(PX*(X(I
    )),PY*(-Y(I)))
740:NEXT I
750:GLCURSOR (P1,(
    P4-90-25*RR)):
    CSIZE 2
751:IF L>16CSIZE 1
    :GLCURSOR (P1,
    (P4-75-25*RR))
753:LPRINT "Y=";
    LPRINT A$(I)
754:IF L>32
    GLCURSOR (P1,(
    P4-90-25*RR)):
    LPRINT A$(0)
760:INPUT "Same ax
    es for next gr
    aph?";A$
770:IF LEFT$ (A$,1
    )="Y"LET RR=RR
    +1
780:IF RR>2BEEP 3:
    PRINT "3 CURVE
     LIMIT":END
790:IF LEFT$ (A$,1
    )="Y"GOTO 120
800:TEXT :LF 5:END


STATUS 1
                2500
```

DIMENSIONS - *SUBROUTINE 1*

Calculating the quantity of A$(n)(nn) for the memory space available is tedious. Re-calculating it for a particular quantity, or particular length of string, is frequently more tedious. For instance, in an INDEX or TEXTHANDLING program, you might want as many variables as possible, with at least a 100; and as long as possible, with a minimum of, say,32 chrs. You can incorporate this subroutine into your program, so as to give it flexibility for this purpose. Alternatively, use it as a utility, when writing the ptogram, and delete after use (thus giving a little more space).It is IMPORTANT that other variables should be dimensioned, and 2-character variables initialised, BEFORE the placing of this subroutine.*

IT IS IMPORTANT TO NOTE that this subroutine does not merely indicate suitable DIMensions: it actually PERFORMS THE OPERATION.

```
10:"D"CLEAR
20:REM  dimension
    other variabl
   es & initialis
   e all 2-chr.va
   riables
30:INPUT "Qty of
   A$(n): max ";A
40:INPUT "Qty  of
    A$(n): min ";
   R
50:INPUT "length
   of strings: ma
   x ";B
60:INPUT "length
   of strings: mi
   n ";P
70:INPUT "priorit
   y: qty(Q)lengt
   h(L)";T$
80:X=(T$="L"):Y=(
   T$="Q")
90:ON ERROR GOTO
   "reduce"
100:"dim"DIM A$(A)
   *B
110:BEEP 1,A,B:
   BEEP 1,B,A:
   WAIT :GOSUB "p
   rint":END
120:"reduce"A=A-
   ABS X:B=B-ABS
   Y:WAIT 0:GOSUB
   "print"
130:IF A<=RAND B<=
   PWAIT :PRINT "
   NO MEMORY!":
   GOSUB "print":
   END
140:IF F=-1GOTO "d
   im"
150:IF A<=ROR B<=P
   LET X=X-1:Y=Y-
   1:F=-1
160:GOTO "dim"
170:"print"PRINT "
   DIM A$(";STR$
   A;")*";STR$ B;
   ": S3-2=";STR$
   (STATUS 3-
   STATUS 2):
   RETURN

STATUS 1
                    539
```

NOTES: This routine may be freely renumbered, owing to the use of labels, as so strenuously advocated by DAVID RIHOY (vol.1,p.103), as directions for GOTO and GOSUB statements.

Where space is a prime consideration, reduce the length of this routine by deleting the REM at line 20. Input prompts could also be abbreviated.

S3-2 *in the display refers to STATUS 3-STATUS 2. This is the free space between the end of your program, and the beginning of DIMensioned and 2-chr.variables.*

FROM THE KEYBOARD

When preparing the 'Index of Titles'(vol.i.p.125) an error was made despite the error-correcting facilities of the INDEX program. The entry under MARKETPLACE read

.....71 72 73.....instead of ...71 73.....

The extraneous 72 was removed by

```
A$(43)=LEFT$(A$(43),20)+RIGHT$(A$(43),20)
```

However this left .....7173....instead of ...71 73.....
It was finally put right by

```
A$(43)=LEFT$(A$(43),20)+" "+RIGHT$(A$(43),20)
```

MAILING LIST - *SUPPLEMENT*

The following addition to "MAILING LIST" (vol.i, p. 120-121) has been found useful in analysing the subscription list. For instance, to discover how many subscriptions outside Europe expire in June 1975 , respond to the prompt 'TEST' by 659 (6 for the month, 5 for the year, 9 for airmail), and to the prompt 'POSITION' by 3 , since the first digit of the digits tested would be the 3rd digit of the subscription number. The result shows how many are in this category, and how many are not.

```
60000:" "CLEAR :
      RESTORE :ON
      ERROR GOTO 6
      0998:WAIT 0:
      TEXT
60005:INPUT "test
      ";T$:INPUT "
      position ";M
      :N=LEN T$
60010:READ A:READ
      B:FOR F=1TO
      B:READ N$:
      NEXT F
60020:A$=STR$ A:IF
      MID$ (A$,M,N
      )=T$LET Q=Q+
      1
60025:IF MID$ (A$,
      M,N)<>T$LET
      U=U+1
60029:PRINT "<>";U
      ;"  =";Q
60030:GOTO 60010
60998:BEEP 5:
      LPRINT T$;"
      <>";U;"  =";
      Q
60999:WAIT :PRINT
      T$;" <>";U;"
        =";Q
```

INDEX PROGRAM - *SUPPLEMENT*

The actual use of the 'INDEX' program (vol.i,p.119) can present problems. As it stands, it is necessary to DIMension the length of strings to fit the largest item, (and this is limited to 80 chrs.). If the majority of items are much smaller, much space is wasted, and the quantity of items DIMensioned is unnecessarily limited. This may be obviated by amending the program, so that when an addition to the 'number' accompaniment to a string would exceed a set length, a new but identical 'title' is created - BUT instructions are given to the effect that , during printout, a title is not printed if it is identical to the previous title, but its 'number' accompaniment follows immediately upon the number accompaniment of the previous string. Some suggested amendments to perform this are listed below. They have not been tested, and some further amendments may be necessary to fit in with your own version of this program.

```
   77: IF LEN B$(G)+LEN B$(N)>72 GOTO 100
 2015: IF A$(K)=A$(K-1) GOTO 2030
```

MINDBOGGLE CORNER

First, a quickie - no prizes. Take a simple program like

```
1: "A" FOR F=1 TO 999: NEXT F
```

Now if you start executing the program by RUN, the prompt > on the left of the display will disappear. But if you start by DEF A the > prompt will remain. But supposing you execute by GOTO 1, what will happen then? And why?

No entries as yet for the January competition. Is it too hard? Or too easy? The prize is not worth winning? Or the stamp not worth wasting? The number of entries received for most competitions is disappointing, and I wonder whether MINDBOGGLE should be changed, or abandoned.

At any rate, here is a puzzle which requires no intellect, just patience and imagination. "To see ourselves as others see us" is usually hard. Use SUPERSKETCH (vol.1, page 99), or any other preferred SKETCH program, to draw a picture of your editor as you imagine him - realistically or symbolically , as you choose. A certain amount of tact is suggested, but not compulsory. Usual prize. Closing date: April 1st. On second thoughts, make that April 2nd.

EASI-ONE! *Software Review*

This remarkable program is six programs in one. Most (but not all) of its features are described in the advertisement on page 11. I particularly like the clever method by which the destruction of data is avoided when you key RUN. A difficult point, however, about the STATS program is this: it produces information, but no graphs, owing to space restrictions, and the author suggests that EASI-ONE is designed for day-to-day applications, whereas for the big applications EASITREND is recommended. But people do not work like this. The expert statistician requires mathematical information, which he is prepared to patiently plot. It is for the immediate day-to-day applications, for the casual or amateur user, that an illustrative graph is required.

Readers have had various reactions to EASI-ONE. J.K.GAUTON found no difficulty in using it, with the manual beside him, though he was disappointed that the CALC section did not produce automatic row and column totals. He writes: *"This program offers practically everything that the average commercial user needs at his fingertips. However I found entry of data in all modes to be too slow, ...and I would have liked a 'sort' facility on the Notepad module. These are personal preferences, and in no way detract from a very well thought-out and useful program. If these are the facilities you need, if you are prpared to study the manual, and are willing to sacrifice speed for accuracy, this a very good buy."* He adds that he had no difficulty in CLOADing the cassette (a point that cannot always be taken for granted with some software).

H-H.HEINE is even more enthusiastic. From Istanbul he writes: *"A few days ago I received MINIMICRO's latest creation EASIONE. In one word - MAGNIFICENT! At least this a program neatly tailored to the capabilities of the PC 1500 with no more intentions to do a job which better should be left to the big brothers. Specially the Spreadsheet module has become very handy due to vastly increased calculating speed and the possibility to insert more complex formulae than in EASICALC."*

The view taken by C.P.UNDERWOOD is (as usual) more critical. He says *"While in many ways this a very good program, I had some difficulty with it. I made a number of mistakes while learning to use it, and found the errors not easy to recover from and return to where I wished to be. It is not as streamlined as one would desire, and in a program of this sort, which is meant to be carried always in the computer, if that is what you need, streamlined use, particularly for the impatient business user, should be a prime consideration. This sort of easy flow takes a lot of programming - I mean not just effort, but actual memory space, which of course in the present instance is just not available in 8K. I think the program is a very brave attempt to be 'all things to all users', but owing to the lack of adequate space in 8K to do all these things as well as they might be done, I found it irritating to try and use. It must be admitted that with 6 programs moulded into 1, there is 6 times as much to learn, which is a further strain on the time and effort the user has available, and perhaps I did not study the manual quite long enough, though I spent all the time on it I could spare. The manual, incidentally, is clear and comprehensive, though the copy I had was not so clearly printed,requiring additional effort, which would better have been directed to the complexities of the program."*

EASI-ONE is developed by MINIMICRO SOFTWARE, and is distributed by ELKAN ELECTRONICS, price £24.95 (VAT incuded)

ELKAN ELECTRONICS

PRESENTS...
A REVOLUTIONARY NEW
PROGRAM FOR THE SHARP
PC-1500 POCKET COMPUTER...

e1 EASI-ONE!

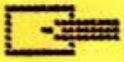
WHAT IS EASI-ONE!

Six programs rolled into ONE! Always in your machine and always available to solve hundreds of problems without the effort of having to load from tape each time! EASI-ONE! incorporates a calendar/alarm clock, an electronic notepad, a unit conversion facility, a simple, but powerful text editor, statistics and forecasting facilities - and an electronic spreadsheet calculator.

Each of the program 'modules' is selected by a single key-press from a graphics-display menu - and each can be cleared when you have finished with it. AND YOU CAN HAVE ALL SIX MODULES SET UP IN YOUR MACHINE AT THE SAME TIME!!

ALARM

Combined with an easy-to-read clock & calendar (automatically displayed on power-up), the ALARM can be set for up to 99 diferent times. At the set times, the alarm will sound and a message (pre-set by you) will be displayed.

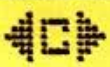


NOTEPAD

Used as a casual scratch pad or an address/telephone index, the NOTEPAD can hold up to 99 lines of 23 characters - about 400 words. NOTEPAD is equipped with a 'Find' facility to locate entries rapidly.

TEXT

A handy text editor with line insert/delete facilities, word carry-over to next line, selective printing of all or part of document and other easy-to-use facilities. TEXT can handle 80 lines of 36 characters (99 with the CE-161 16K module) - nearly 500 words. Can link in with NOTEPAD to transfer NOTEPAD lines to TEXT (names, addresses etc.) Ideal for memos, informal letters etc., using the CE-150 printer.

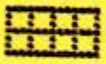


CONVERT

A handy unit conversion facility always at your fingertips. Miles to kilometres, dollars to yen - enter your own choices (up to 99 of them) and CONVERT will always be there for instant use.

STATS

Rapidly calculates total, average & standard deviation for each of two columns of data that you enter. Using linear regression techniques, STATS can also attempt to find a relationship between the two sets of data (with linear, exponential or root relationships) which can then be used for forecasting.



CALC

An electronic spreadsheet calculator - ideal for solving the 'what-if' type of problem. Up to 110 cells available (over 400 with the 16K module). Can accept complex formulae up to 16 characters long (e.g. B5*(A5-A2^12)). Can handle Boolean logic equations. Uses machine code routines for speed. Data from either rows or columns can be moved over to STATS for analysis.

Other features in each module include:- 'browse' facility, hard-copy printout, rapid movement to beginning or end of module & numeric rounding of displayed figures.

What equipment does EASI-ONE! need?

- Sharp PC-1500 or Tandy PC-2 Pocket computer.
- Either 8K or 16K RAM module.
- CE-150 Printer/Cassette interface.
- Compatible cassette recorder.

Designed for Pocket Computers!

EASI-ONE! is no look-alike program adapted from other micros. It is specifically designed for the pocket computer with its ability to retain programs and data. For the big applications, you'll want to use the other programs in the EASI-range - EASI-CALC, EASI-TREND, EASI-FILE and EASI-CASH. But for all those day-to-day applications, EASI-ONE! is the ONE!.

WHAT'S IN THE EASI-ONE! PACKAGE?

EASI-ONE! comes to you in the form of a data-quality program tape and a 50-page user's manual with a rapid reference section. EASI-ONE! costs £24.95 (incl. VAT) and is available from your local dealer or direct from the sole UK distributors:

Elkan Electronics, FREEPOST (no stamp required), 11 Bury New Road, Prestwich, Manchester M25 6LZ

Telephone 061-798 7613 (24-hour service)