VOL.2
NO.12

STATUS 1500

JANUARY
1985

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

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

ANOTHER YEAR

We come to the end of Volume 2. I hope you have enjoyed it as much as I have, and that it has been as useful to you as it has to me. Let me take this opportunity to express my thanks to all those who have contributed, including those whose contributions have not found space to appear. A few readers, who have moved on to more conventional machines, have had the kindness to write in and say that they will not be renewing. To you also let me express my appreciation of your support hitherto.

Most of you however will (I trust) renew your subscriptions for Volume 3. Last year renewals ran at 83.4% of the list, and I hope that we shall do almost as well this year. You will be pleased to hear that - despite inflation - there will be no change in the cost of subscription. This will not be changed this year. However a threatened sharp rise in the price of paper, and therefore of printing prices, may make it necessary to cut costs: and so we might be obliged to cut down from 12 issues per year to 11, or even possibly 10. The August issue could be the first to go: you would be warned well in advance. Bound volumes of Volume 2 will be available as soon as the Subject Index is compiled, probably in a month or two. The price of bound volumes is raised to the same price as annual subscriptions, including postage, with effect from February 1st.

But is it worth renewing? Or have we exhausted the potential of the PC 1500? By no means exhausted yet! Indeed, we already have some splendid and original material to hand. A series of articles by Frank Odds explains how to reprogam your keyboard entirely. Have you ever run short of DEF labels, and wished you could use the QWERTY row? Now you can. Have you ever wished to reassign the commands this row contains? Now you can do so. What else? A series of "Machine-Code Notes", starting in the spring, will explore this aspect which has been somewhat neglected of late, and should furnish some neat subroutines of use to everyone. For beginners (including perpetual beginners) a new series entitled "Second Steps!" will re-examine some of the trickier basic techniques involved in handling the PC 1500.

Stay with us for 1985!

SIGNALS

A.E.L.COX confesses that he finds that the Shell Sort is not actually the fastest form of Sort: there is another called a Quicksort which is even faster. However the routine is lengthy, and therefore unsuitable for the PC 1500; but he hopes to develop an abbreviated adaptation for our machine. He also confesses that his Shell Sort is not really a Shell Sort at all, since the genuine article uses a quite different selection of variables for comparison.

A disappointment. I would have liked to know what a Shell Sort really is. Nevertheless I am sure that the routines you gave can be useful.

F.C.ODDS signals that recently he had to sort some 1500 items. He used a simple Bubble Sort, and the program took 32 hours to run!

It is a pity you were unable to adapt for your purpose any of the routines we have printed. We did our best!

SYDNEY LENSSEN is disappointed that the random noises which accompanied the program on the cover of the Xmas number did not sort themselves out into a Xmas carol.

Now is your chance! Next Xmas Mindboggle competition will be for a Xmas carol, or Advent calendar (or both). Why not start working on this now?

IAN TRAYNOR wonders whether any readers have used the PC 1500 for communications, via CE 158 and modem.

Except for R.MILLEN's access to REWTEL, (see vol.1, page 16) I have not been informed of any such attempts. I would be very interested to receive any information on the subject.

IAN TRAYNOR also mentions that he was amused by the reference to dinosaurs by T.RASMUSSEN (page 99) since the computer he has chosen, the Casio FP200, is itself something of a dinosaur, almost obsolete. He adds that the PC 1500 is faster than the 2 new SHARP Pocket Computers and the 2 new Casios.

C.P.UNDERWOOD points out that in the Ghost Program (page 101) you need not rehabilitate the program area by NEW 0. To restore the program, key: POKE 30821,PEEK 30819,197, PEEK 308219+1,191,PEEK 30819,197 and POKE X+169,50,27. However he is puzzled by the function of the expression in line 30: POKE X+000,00. He wonders what this does.

Nothing at all. When devising this program I found it easier to structure it first, and fill in the details afterwards. Otherwise every time I inserted the figures to POKE, for concealing lines, the actual insertion would have altered the figures I needed to POKE, and I might have found myself 'flying in ever-decreasing circles'.

ANGUS CRAWFORD regrets that a program I sent him, recorded with the TOOL-2, failed to FLOAD with his module. He presumes that these modules vary too much in their individual characteristics.

It could be just my machine. Has anyone succeeded in loading materiel saved on a different module?

DAVID RIHOY corrects my interpretation of 'Phasencodierung' as 'Analogue' (see page 38). He says that 'Phase Encoding' is a recognised system, whereby the 8 bits, together with a 9th check bit, are recorded in parallel, instead of serially.

Perhaps the implications of this contribute towards the difficulty described above. It may be that the variation is not in the modules, but in the recording systems used, and the slightest variation in head alignment makes transfer impossible.

PEEK POKE & MEMORY - XXII

a program reader

This program was written for a new subscriber who is blind. Since it is laboriously slow to use, it is fortunate that he is not likely to need it, because he has a very wonderful instrument called an OPTAKON which translates visual images, even the display of this computer, into sensory impulses. Nevertheless I print the program, because the PEEK techniques used are interesting, and will repay study, particularly for those who have difficulty with the concepts of ASC-CHR$-STR$, on which the program is based.

It is presumed that the user can find his way around the ASCII codes of the main characters, numerals, and punctuation marks: but does not have access to a complete list of Reserved Word codes. The program in action is less difficult than its exposition on this page. It is to be MERGEd with the subject program that is to be examined. You are invited to INPUT a line number, and a speed (1 to 5 suggested). After reading each character, key ENTER for the next one. The basic principle used is to represent each character numerically, (as you would get if you PEEKed the location it was in), and to represent it by clicks for each digit of that number. Since more than 5 clicks are hard to count at speed, and waste time too, digits over 5 are represented by double-clicks for the excess over 5. Thus a colon, CHR$ 58 would be represented as 058, to make up the 3 possible digits. You would hear BEEP (for 0), 5 clicks (for 5), and 3 double-clicks (for 8). Where you hear double clicks, add 5 to their total to make up the digit they represent. The letter C, for instance, would give BEEP - - 1 doubleclick - 2 doubleclicks. When it comes to Reserved Words, starting 240, 241, etc., the digits are read, and then the program retrieves an abbreviation from the table, and gives the digits of the 3 characters of the abbreviation. Space does not permit printing the whole table here. You can easily write your own, if you have the time, using the table on page 28 of volume 1. It is not necessary for this to be complete. If you make up your own table, note that the number label (i.e. "24180") is NOT expanded to 6 digits: but the 3-chr. label MUST be 3 characters, by adding a space if necessary, as in line 63058. What has happened is that the program has recognised a Reserved Word, in line 61050, has taken the PEEKs of its 2 locations, transformed these 2 numbers into a string, and searched for a label identical with that string, by the expression GOSUB Q$. I should add that initially the first 3 bytes of the line, containing the line number and length, are read without translating out of 256ary. Although you must specify the start line for the operation, you are not limited to that line, and may continue without further intervention.

I doubt whether any of you have a real need for this program, other than for study - although if your printer and your display both were to fail simultaneously you could no doubt make use of it. However if anyone does have a real need, I would be happy to send a cassette containing the program and the full table. Note that a number of abbreviations are not the official keyboard ones, owing to the necessity of sticking rigidly to 3 characters.

Other operations:

DEF X allows you to INPUT a string, and read it by DEF C.
DEF D allows you to read the display.
DEF K will read any key pressed. You may find this too slow.
DEF M reads the type of ERROR committed.

```
61000 "A"POKE 30825,PEEK 30821,PEEK 30822
61001 WAIT :V=3:GOSUB 61900:INPUT "SPEED?";V:REM ?GOSUB "D"
61002 GOSUB 61900:INPUT "LINE?";N
61004 ON ERROR GOTO 61800
61006 RESTORE N:P=256*PEEK 30886+PEEK 30887-3
61010 F=PEEK P:F$=STR$ F
61030 GOSUB 61600
61040 ON ERROR GOTO 61150
61050 Q=PEEK (P-1):IF Q=230OR Q=231OR Q=240OR Q=241LET Q$=STR$ Q+STR$ F:GOSUB Q$
:GOSUB 61500
61150 P=P+1:GOTO 61010
61500 BEEP 1,17,1717:FOR G=1TO 3:F$=STR$ ASC MID$ (G$,G,1):GOSUB 61600:NEXT G
61510 RETURN
61600 IF LEN F$<3LET F$="Ø"+F$:GOTO 61600
61602 FOR J=1TO 3:@(J)=VAL MID$ (F$,J,1):NEXT J
61605 X=(A=0):Y=(B=0):Z=(C=0):R=(A>5):S=(B>5):T=(C>5)
61610 FOR K=1TO A+X-5*R:BEEP 1+R,7+30*X+9*R,7+43*X:BEEP 1+V,0,0:NEXT K
61615 BEEP 5+3*V,0,0
61620 FOR K=1TO B+Y-5*S:BEEP 1+S,7+30*Y+9*S,7+43*Y:BEEP 1+V,0;0:NEXT K
61625 BEEP 5+3*V,0,0
61630 FOR K=1TO C+Z-5*T:BEEP 1+T,7+30*Z+9*T,7+43*Z:BEEP 1+V,0,0:NEXT K
61640 CLS :PRINT CHR$ VAL F$;"/";F$;F
61650 RETURN
61800 BEEP 1,1,1:N=N+1:GOTO 61006
61900 BEEP 1,37:BEEP 1,14:RETURN
62000 "X"ON ERROR GOTO 62010
62005 DIM X$(0)*80
62010 GOSUB 61900:BEEP 7,14:STOP :REM  key> X$(0)=?$(?) or STR$ n, etc.
62020 "C"FOR L=1TO LEN X$(0):F$=STR$ ASC MID$ (X$(0),L,1):GOSUB 61600:NEXT L:END

62500 "M"BEEP 2,255,200:F$=STR$ PEEK 30875:GOSUB 61600
62505 BEEP 1,255,200
62510 E=256*PEEK 30900+PEEK 30901:E$=STR$ E
62520 FOR H=1TO LEN E$:F$=STR$ ASC MID$ (E$,H,1):GOSUB 61600:NEXT H:END
62600 "K"WAIT 100:F$=STR$ ASC INKEY$ :F=0:GOSUB 61600:GOTO "K"
62700 "D"D=0
62710 F$=STR$ PEEK (&7BB0+D):GOSUB 61600
62715 ON ERROR GOTO 62740
62720 IF F$="013"RETURN
62730 D=D+1:GOTO 62710
62740 BEEP 5,37:END


63000 "230128"G$="CSI":RETURN
63002 "230129"G$="GRA":RETURN
63004 "230130"G$="GLC":RETURN
63006 "230131"G$="LCU":RETURN
63008 "230132"G$="SOR":RETURN
63010 "230133"G$="ROT":RETURN

63054 "240188"G$="TES":RETURN
63056 "24180"G$="AND":RETURN
63058 "24181"G$="OR ":RETURN
63060 "24188"G$="MEM":RETURN
63062 "24191"G$="TIM":RETURN
63064 "24192"G$="INK":RETURN
63066 "24193"G$="PI ":RETURN
63068 "24196"G$="ASC":RETURN

63154 "241150"G$="IF ":RETURN
63156 "241152"G$="LET":RETURN
63158 "241153"G$="RET":RETURN
63160 "241154"G$="NEX":RETURN
```

ADDING MACHINES by Mike O'Regan
a diversion

On my office wall I have a SOROBAN (Japanese 5-bead abacus) labelled "to be used in case of COMART CP520 breakdown". In my desk drawer I have a chunky brass leather-cased (and practically indestructible) ADDIATOR. This is a sort of pocket abacus which served me well for years before the introduction of the electronic calculator. It still gets occasional use.

There is still a need for the adding-machine-cum-calculator. Our cost-accounts department has one on every desk, each with a well-worn "+" key which shows that it is used almost exclusively for adding up figures.

A couple of years ago I had a HEWLETT-PACKARD 41c Programmable Calculator which had the desirable attribute of a PAUSE command during which the keyboard remained "live" (able to accept input directly). This enabled me to write a simple automatic adding program which would keep a running total of entries and allow unlimited input during the (BEEP signalled) PAUSE without the need to key in "+" or ENTER. I found that with a little practice I could use the keyboard without looking at it and, with the calculator forcing the pace, I was able to halve my time for adding a column of figures.

The program below (PiCoCOMP) is a version of that program, modified because the PC 1500 lacks a live keyboard PAUSE. It will:

Allcw the number of decimal places to be set from the keyboard.
Pause for sufficient time to accept any numerical entry.
Display each digit as it is entered.
Keep a record of each entry.
By pressing a single key:
  Display each entry, serially numbered, and the total (Key P).
  Produce a justified printout of each entry, and the total (Key L).

```
 10 " "CLEAR :CLS :DIM N(250):C=0
 20 INPUT "Decimal places ";D:IF D=0USING :GOTO 40
 30 FOR K=1TO D:D$=D$+"#":NEXT K:D$="######."+D$:USING D$
 40 T=TIME :WAIT 0
 50 A$=INKEY$
 60 IF TIME >T+.0001THEN 120
 70 IF A$=""THEN 50
 80 IF A$="L"THEN 140
 90 IF A$="P"THEN 200
100 BEEP 1:B$=B$+A$:PRINT B$:GOTO 40
120 N(C)=VAL B$:S=S+N(C):IF B$<>""LET C=C+1
130 B$="":CLS :BEEP 1,20:GOTO 40
140 FOR K=0TO C:IF N(K)=0THEN 170
160 LPRINT N(K)
170 NEXT K
180 GRAPH :LINE -(220,0):TEXT
190 LF 2:LPRINT S:LF 4:CLEAR :END
200 FOR K=1TO C-1
210 PAUSE K;"  ";N(K):NEXT K:WAIT :PRINT "Total: ";S:GOTO 40

                                          STATUS 1 = 431
```

EASTER
Easter Sunday for any year

by E.Macmillan

```
160 INPUT "YEAR FOR EASTER",Y:G=Y
180 IF G<19GOTO 210
190 G=G-19
200 GOTO 180
210 G=G+1:C=INT (Y/100)+1:X=INT (C*3/4)-12
240 Z=INT ((8*C+5)/25)-5:D=INT (5*Y/4)-X-10:E=11*G+20+Z-X
270 IF Y<1753LET E=E+8
280 IF E<0LET E=E+30
290 IF E<30THEN 320
300 E=E-30:GOTO 290
320 IF E=24THEN 350
330 IF E=25AND G<11GOTO 350
340 GOTO 360
350 E=E+1
360 N=44-E
370 IF N<21LET N=N+30
380 B=D+N
390 IF B<7GOTO 420
400 B=B-7:GOTO 390
420 N=N+7-B:IF Y<1753LET N=N+3
440 IF N>31GOTO 470
450 PRINT "EASTER DAY MARCH";N-31;Y:END
470 PRINT "EASTER DAY APRIL";N-31;Y:END
```

IMPROVEMENTS

title: CHARACTER CONSTRUCTOR (page 81)
improved by: C.P.UNDERWOOD
purpose: Allows up to 48 lines, referenced.
Allows amendment of individual lines by DEF A.

```
 5 CLEAR :TEXT :CSIZE 2
10 POKE STATUS 2+2,253,168,72,122,74,9,88,119,90,208,190,167,136,154
20 DIM X(47),A(47),B(47),C(47):F=0:WAIT 0
30 GOSUB 80:GOTO 40
40 BEEP 1:X(F)=64*A(F)+8*B(F)+C(F)
45 IF F<16LET X$=X$+CHR$ X(F):GOTO 60
50 IF F<36LET Y$=Y$+CHR$ X(F):GOTO 60
55 Z$=Z$+CHR$ X(F)
60 GOSUB 98
65 IF A(F)=3GOTO 75
70 F=F+1:GOTO 30
75 TEXT :LF 7:FOR D=2TO 6STEP 2:LCURSOR D-1:CSIZE D:CALL STATUS 2+2
76 NEXT D:END
80 GOSUB 85:INPUT "control ";A(F):GOSUB 85:INPUT "direction ";B(F)
81 GOSUB 85:INPUT "length ";C(F):RETURN
85 CLS :PRINT F;">  ";:RETURN
90 "A"INPUT "line to amend:";F:GOSUB 80:X(F)=64*A(F)+8*B(F)+C(F)
91 POKE &77D0+F,X(F):TEXT :CSIZE 2:LF 2:GOSUB 98
95 INPUT "MORE?(Y/N)";J$:GOTO (90-15*(J$="N"))
98 LPRINT F;">";:LCURSOR 4:LPRINT A(F);B(F);C(F);" =";
99 LCURSOR 12:USING "####":LPRINT X(F):USING :RETURN
```

LETS WRITE A PROGRAM - concluded

The series is now concluded, although the program is not. Several subroutines remain to be written, but I do not wish to carry this project over into another volume, particularly since there has been no feedback at all, which scarcely betokens much interest on your part. This was originally intended to be a workshop project, with your participation. The absence of this makes one of the objects of the exercise impossible to pursue: there are no rival approaches to compare.

Despite the disappointment, there is still much to learn from the series. You may not think so, but perhaps you have absorbed some of its lessons unconsciously. Recently we have been bogged down in the development of specific routines; and these, however interesting and challenging, were not the primary object. This object was: the approach to constructing a program. This was too successful! and so nothing but the individual routines remained to be done. This is the first, middle and last lesson: THINK WHAT YOU ARE GOING TO DO BEFORE YOU START DOING IT. We spent the first few months working out our intentions as deeply as possible, and the result was that all the routines we developed slotted neatly into place. It will be quite easy to position the remaining ones, if you care to write them for yourself. I do not intend to be more specific than this. All the hints and clues you need have already been given. If you think otherwise, write in and say so, in detail.

In a way, we were fortunate in our choice of subject. The problems, and the required results, existed before we started. We were careful to spend much time and effort planning before we wrote a line of code, and this paid off. But with many programs the original concept is less precise. It is not until one has written a fairly simple program, possibly with an ingenious trick or two, that inspiration for further expansion descends; and from these expansions come yet further ideas. Ideally, at this stage, one should use what one has written so far as raw material: replan, and rewrite the entire program. But who has time to do so? An example of a program which has grown by accretion is PROGRAM READER on page 116. Although some of the tricks are recommended for study, the layout is such that even its author is puzzled to follow its flow. For an example of a well-designed program, (as far as arrangement is concerned), see TONY WILLMAN's SPACE LANDER on page 104. Here the various blocks are each entirely self-contained,.

The layout of the Roulette program was particularly important for a reason that I did not appreciate at first. Whenever I added a new routine, it may have been a month or even two since I last looked at the program. It was only the clarity and flexibility of the original scheme that allowed me again and again to take up where I had left off.

So let me emphasise one last time this message:

PLAN YOUR PROGRAM BEFORE YOU START WRITING IT!

MINDBOGGLE CORNER

No competition this month, but the deadline for the self-portrait competition is extended to 29th January. Please have a go!

Only 2 entries for the ROM CHR$ contest. DAVID RIHOY submitted a Xmas tree, but this, although ingenious, was too large to actually use as a character. C.P.UNDERWOOD also submitted a tree, and this is bespattered all over the front page of the Xmas number. The coding is:

9,121,65,73,99,81,67,113,25,83,122,1,91,17,123,25,17,91,1,107,49,41,75,49,107,1,74,243

INDEX OF TITLES

INDEX OF NAMES

INDEX OF SUBJECTS

[This Index does not duplicate the INDEX OF TITLES, and therefore should be used in conjunction with it. Cross-references below are indifferently to both indexes]