VOL.2
NO. 2

STATUS 1500

MARCH
1984

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

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

LOBSTER SOUP

I had long determined that on the day when I got a printer, I would sit in a deck-chair in the park, tapping out an editorial for the newsletter: on my return home I would connect computer to printer, and watch the text being printed out. Alas! it is late on a cold wet February evening, and relaxing in a deck-chair in the park is just "not on". So I have settled instead for a celebratory supper, and I am keying in this text between courses.

The printer is a JUKI 6100 daisywheel, parallel interface, and it was chosen after very long deliberation. I was attracted by parallel interfacing, which seemed less complex to operate than serial - though this meant I was unable to take advantage of the cheap interfaces at TANDY, since these are RS232 only. I liked also the wide choice of typefaces; and was influenced by the fact that CORTEX COMPUTERS, from whom I purchased it, were really concerned to make sure that I would have a viable combination of printer and computer.

The printer should make editing the newsletter much easier. Listings should now be more legible; and infinitely greater control of spacing should make it tidier - as well as permitting greater flexibility in the length of text on each page. However during the changeover period it may be necessary for you to tolerate an even more bewildering variety of typefaces and layouts than usual, until all is sorted out.

Perhaps I may add that although I have only had the printer a few hours, it is already chattering away merrily. The 2K buffer helps to cope with some of the slower features of SUPERTEXT, which required an additional few hundred bytes of program to adapt to this printer. My initial over-hasty attempt was slightly ludicrous since it stripped every x y and z from the text. However things now appear to be under control. - And so my banquet is over. Home now to see this article printed out. Subscriptions will not immediately be raised to pay for the lobster soup. On the other hand, I have developed an insatiable appetite for exotic daisy wheels...

12 LOBSTER SOUP
13 SIGNALS
14 PEEK POKE & MEMORY - XIII
14 LEAP YEAR
15 LETS WRITE A PROGRAM - II
16 MINICALC
17 CARD SHUFFLE - Subroutine 2
18 ROM INFORMATION
18 FROM THE KEYBOARD
18 MINDBOGGLE CORNER
19 "GETTING STARTED" - book review
20 MARKETPLACE

T.GREEN works in finance and insurance, and would like to see more programs on these subjects.

It is not the purpose of the newsletter to publish the type of programs that are easily available commercially, such as in TANDY's Business Finance Pack, or in the many books on financial programs in BASIC, which need little adaptation. However we have a number of subscribers involved in insurance: if they care to submit programs on this subject, I shall be keen to print them - provided these are not too technical for the intelligent layman to understand.

Several readers have had difficulty with the GRAPH program on page 7, owing to poor reproduction, and some confusion between characters.

Usually a little effort and commonsense - not always available - will clearly indicate the answers. However if any reader has had insuperable difficulty in copying this excellent and useful program, please send a large SAE, and you will receive a legible transcription.

SIMON COX is no longer a subscriber to this newsletter, and no further letters can be forwarded to him with regard to his claimed 96K.

Letters from one subscriber to another can always be forwarded, provided a stamped envelope is supplied, but it is obviously impossible to compel the recipients to answer; nor will the Editor accept responsibility for the results of such correspondence.

IAN TRAYNOR is disappointed with the new series 'LETS WRITE A PROGRAM' He thinks that the absence of hard facts will make it of little use.

This is a deliberate misunderstanding of the purpose of the series. There are many books and articles which give the hard facts. The great difficulty lies in selecting, and making use of, these facts. The particular concern of this series is the mental approach necessary before starting to write a large program; and the difficulties encountered as it progresses.

DAVID BOWRING recently damaged the 'System Cassette' for his CE 153 Software Board. This could have been disastrous, since he uses this board to control machine-tools, and SHARP of course are notoriously unhelpful. Fortunately 2 subscribers were able to supply copies, and he is now operational again.

Most grateful to F.C.ODDS and C.A.F.LEDSAM for their very kind help.

M.GREENING-LEWIS suggests that a regular feature on 'Improvements' to programs would be useful and popular.

We have recently printed several such items, though not under that heading. We will experiment with your idea as an occasional feature. However such 'Improvements' must be either to programs previously printed in the newsletter, or to routines in common use. They must be substantially more useful than the originals, not just variations.

GUY BURTON lives in a remote part of Wales, without electricity, and therefore cannot recharge his CE 150. He uses a 9-volt battery, connected externally, Owing to the device requiring constant amperage, he had planned to interpose a circuit to maintain this, but so far this has not proved necessary.

Most interesting. This seems a very useful system, since the printer will work for a maximum of 3 hours without recharging, in normal use, and when away from a power-source it is annoying to run out of power. Nor is it always convenient to fish around under other peoples' desks for a socket to plug into. The connections seem simple, but are beyond my capabilities. Have you ever thought of marketing such a device? I would be your first customer.

PEEK POKE & MEMORY - *XII*

Yet More MAILING LIST - A Use For PEEK & POKE

Some of you may wonder why this month your subscription number has changed from 27158... to 20158... , or some such change. When new subscriptions or late renewals arrive just while I am preparing to mail the newsletter, it is not quite convenient to update the subscription list at that moment: and there is always a danger that the new arrivals might get overlooked, or duplicated. So, temporarily, I update them, with the figure 7 as the 2nd digit. When I have despatched the main batch, I do a supplementary batch for all numbers with 7 in second place. The labels for these are done automatically, using a slight modification of the routine contained at line 10025 and lines 16000 to 16100 (vol.1, p.121), and described on the page previous to that.

When the supplementary batch is done, the figure 7 must be restored to 0, for normal use. It is tedious to do this by hand, and one could get missed. So I have veryquickly developed the routine below, to examine all subscription numbers with 7 in the 2nd place, and POKE 0 in place of the 7.

Line 61000: *..F=00256... so that I can change this number without altering the length of the program.*

Line 61005: *shows the line number being worked on, and the total of changes made.*

Line 61010: *...IF PEEK F=141... i.e.if the first statement in the line is DATA ....AND PEEK(F+1)<>34... this was essential., to confine the operation to DATA statements followed by a number. (34 is the code for inverted commas). An original version omitting this statement changed all street numbers starting with 7 !*

Lines 61020 and 61700 to 61720:
speeds up the program. Since all line numbers holding subscription numbers are multiples of 10 , with addresses on succeeding lines, only line numbers ending in 0 are examined: otherwise the line is bypassed, by adding to the loop-counter F the contents of the byte holding the line-length The precautionary statement about inverted commas in line 61010 is probably now redundant. (For the way line numbers and line-lengths are formatted see PEEK POKE & MEMORY - II & III, pages 13 and 24 of volume 1).

```
 61000 "B"FOR F=00256TO STATUS 2:WAIT 0
 61005 IF PEEK F=13PRINT (256*PEEK (F+1)+PEEK (F+2));P
 61010 IF PEEK F=141AND PEEK (F+1)<>34AND PEEK (F+2)=55POKE
  (F+2),48:BEEP 1,15,500:P=P+1
 61020 IF PEEK F=13GOSUB 61700
 61030 NEXT F:END
 61700 N2=256*PEEK (F+1)+PEEK (F+2)
 61710 IF N2/10-INT (N2/10)<>0LET F=F+2+PEEK (F+3)
 61720 RETURN
```

LEAP YEAR - FROM THE KEYBOARD

February 29th will put the date element in TIME 1 day out. If you have not already reset your TIME, there is no need to do so completely. Merely key:

TIME=TIME-100 [ENTER]

If you have already re-entered TIME, you will need to remember this advice for another 4 years, before you can make use of it. However it can of course be adapted to any other minor adjustments to the TIME.

LETS WRITE A PROGRAM - *II*

Writing a program is not really quite such an elaborate process as this series of articles would seem to suggest. Many of the sort of decisions we shall discuss are normally taken unconsciously, or , at any rate, in that state of semi-consciousness in which many programmers seem permanently to exist. Many of these decisions, again, are not exactly programming decisions, but really concern the objective, which is more usually clearly defired beforehand. But with our limited resources, both from the technical and the intellectual point of view, there is interaction between objective and method: we must be careful not to define our objective so rigidly that we find at some point our resources are not adequate for the essential task.

Anyway, lets start with a rough outline of what we intend to do. In my own case this usually is a few pages of almost illegible notes: for the sake of these articles it must be set down more precisely than I would normally do.(Here I may mention a habit of my own which I find useful: to give the various headings labels from the Greek alphabet, to avoid any possibility of mistaking them for DEF labels, or numerical sequence. This comes later.)

So here are the provisional main headings, for our game of ROULETTE. Note that most divide initially into 2 sections: problem of calculation, and problem of display.

a) Initialisation (DIMensioning etc)

b) How many players?
b.2) INPUT names or initials of players,
and their capital.
b.3) BANK's capital: limited or unlimited?
b.3a) Display bank's capital?

c) 1st players name displayed "PLACE YOUR BETS! (name)"
c.1) 1st player backs :: even chances
:: columns/dozens
:: groups
:: single numbers
(players remaining capital displayed after each bet?)
c.999) repeat successively for each player. If finished.....

d) SPIN WHEEL : Winning number displayed. Or numbers displayed sequentially, slowing down and finally stopping at winning number? (??? if so, in numerical order? or wheel order)?
d.2) When calculating winning number, also calculate AND DISPLAY: "RED or BLACK" (There is in fact a formula).

e) Recalculate each players remaining capital according to his winnings or losses on this spin. Display or printout. Recalculate bank's wins or losses; adjust; display.
e.999) When done, return to c) above.

a.2) Graphics: draw roulette TABLE, showing columns, and whether numbers are RED or BLACK.

z) Provision for SYSTEM betting, with single initial INPUT?

PROBLEM: If number of players is not limited, and the number of bets each can make is not limited, we are going to need a terrible lot of DIMensioning to hold every bet: and patience while all results are calculated. There is in fact an answer to this; see this column next month!

MINICALC

by J.K.Gauton

This simple CALC program allows the entry of cells in columns and rows: and gives the totals of columns, the totals of rows, and the Grand Total. Values are entered serially in columns. If you had 2 rows, and 3 columns then **a1, a2** would compose your 1st column, and **a3** would be that column total. Column 2 would be **a4, a5: a6** would be the total of the 2nd column. After the last column total the row totals are given (in this case **a10, a 11): a12** is the Grand Total.

DEF L prints out the values serially.
When using the program without the CE 150 ,alter line 655. LPRINT would be changed to PRINT, and the TAB instructions would be left out.

DEF X permits the changing of individual cell values: these are specified by column and row.The present value of the cell is shown. Key ENTER, and respond to the ensuing prompt with the new value.

```
 10 CLEAR :WAIT 0:CLS                                    [STATUS 1=1134]
 20 INPUT "no. of rows? ";E:F=E:M=F+1
 30 INPUT "no. of cols.?";C:G=C:A=14
 40 DIM A(14+M*(C+1))
 50 B=A+E:A(B)=0
 51 IF G=1THEN 60
 55 D=A+C*M
 60 FOR A=ATO A+E-1
 64 A$="a("+STR$ (A-13)+")"
 65 CLS :PRINT A$;
 70 INPUT A(A)
 72 A(B)=A(B)+A(A)
 75 IF G=1THEN 100
 80 A(D)=A(D)+A(A):D=D+1
100 NEXT A
105 WAIT :CLS
106 A$="COL.TOT.a("+STR$ (B-13)+")"
110 PRINT A$,A(B)
111 IF G=1THEN 171
114 A(D)=A(D)+A(B)
115 C=C-1:E=F:A=A+2:D=D-E+1
140 IF C=0THEN 160
144 WAIT 0
145 GOTO 50
160 PAUSE "ROW TOTALS:"
162 WAIT :CLS
164 FOR D=14+G*MTO D+F
165 A$="ROW TOT.a("+STR$ (D-13)+")"
166 PRINT A$,A(D):CLS :NEXT D
171 PRINT "END":END
580 "X"IF G=1LET L=1:GOTO 600
590 INPUT "col.no.?";L
600 H=14+(L-1)*M
610 INPUT "row no?";R:K=H+(R-1)
615 PRINT "VALUE NOW: ";A(K)
620 B=H+F:A(B)=A(B)-A(K):D=14+(R-1)+G*M:A(D)=A(D)-A(K):J=14+M*(G+1)-1
630 A(J)=A(J)-A(K)
640 INPUT "new value?";A(K):A(B)=A(B)+A(K):A(D)=A(D)+A(K):A(J)=A(J)+A(K)
641 IF G=1LET J=B
642 PRINT "NEW G.TOT.:";A(J)
644 GOTO 171
646 "L"H=14+M*(G+1)-1
648 IF G=1LET H=14+F
650 FOR A=14TO H
655 A$="a("+STR$ (A-13)+")":LPRINT TAB 1;A$;TAB 6;A(A):NEXT A
660 GOTO 171
```

CARD SHUFFLE Subroutine 2

This ingenious routine is extracted from DRAW POKER, by F.C.ODDS. The game itself will be published later this year. It is normally much faster than the system used in BINGO (vol.1, p.111) which compares every Random Number with all the others already picked, in order to avoid duplication. Instead, it chooses a set of locations to correspond with the 52 cards. Then picking a Random Number to correspond with the order in which each card is dealt, it POKES that order number into the location corresponding to the card. If the location is occupied, it simply tries again. Printed here is the heart of the routine: a little effort is required in order to translate card numbers into suits and honours.

```
900 DIM A$(3)*39
905 Z=STATUS 2+2
910 WAIT 0:PRINT " pack being shuffled..."
915 FOR K=1TO 52:POKE Z+K,0:NEXT K
920 FOR K=1TO 52
925 BEEP RND 3,7,5
930 X=RND 52
935 IF PEEK (Z+X)  GOTO 930
940 POKE Z+X,K
945 NEXT K
950 FOR F=Z+1TO Z+52
955 G=INT ((PEEK F-1)/13)
960 A$(G)=A$(G)+STR$ (F-Z)+CHR$ 32
965 NEXT F
970 FOR H=0TO 3
975 LPRINT A$(H)
980 NEXT H
```

You absolutely insist on real hands of cards, properly sorted? Very well. Rewrite the program after line 945. (X is used for 10)

```
 947 CLS :BEEP 1:PRINT "dealing.."
 950 FOR F=Z+1TO Z+52:BEEP 1,11,11
 955 G=INT ((PEEK F-1)/13):C=F-Z
 960 S=INT ((C-1)/13):GOSUB 1000+S
 965 C=C-13*((C>13)+(C>26)+(C>39))+1:C$=STR$ C
 970 IF C>9GOSUB 1000+C
 975 A$(G)=A$(G)+S$+C$+CHR$ 32
 980 NEXT F
 985 FOR D=0TO 3:LPRINT A$(D):NEXT D:END
1000 S$="S":RETURN
1001 S$="H":RETURN
1002 S$="D":RETURN
1003 S$="C":RETURN
1010 C$="X":RETURN
1011 C$="J":RETURN
1012 C$="Q":RETURN
1013 C$="K":RETURN
1014 C$="A":RETURN
```

Sample deal:

```
S2 S6 S8 S9 SK SA H8 HA D4 DX DK C2 CK
S4 SX SJ SQ H2 H3 H6 D2 D5 DJ DA C3 C8
S3 S5 HJ HQ D7 D9 DQ C4 C5 C7 C9 CX CA
S7 H4 H5 H7 H9 HX HK D3 D6 D8 C6 CJ CQ
```

ROM INFORMATION

Only in New Zealand has one of the EASI- programs failed to work satisfactorily. IAN TRAYNOR has discovered the cause. In the later models of the PC 1500 a slightly changed ROM gives different results under specific conditions. These can result in a different residual value for the Loop Counter after the loop is over, and in differing results after the expression IF X THEN.... when X is negative. However it is always possible to amend a program to take care of these variations, once you know. For instance, IF X THEN ...can be amended to IF X>0 THEN ... Here is a chart which shows the effects.

	PEEK	PEEK	value of X	value of X	IF X THEN...
	&C443	&C5BD	FOR X=1TO 3	FOR X=0TO 5	value of X
				STEP 2	-1 0 1
ROM A01	56	129	1 2 3 [3]	0 2 4 6 [6]	No No Yes
A03	59	129	1 2 3 [3]	0 2 4 [6]	No No Yes
A04	59	74	1 2 3 [4]	0 2 4 [6]	Yes No Yes

[The figures in square brackets show the value of X after the loop]

FROM THE KEYBOARD

Suppose you need a line of exactly 40 Xs (perhaps for an EVAL routine). It can be tedious and confusing counting them, owing to the limited display. Here is an easy way. First , key DIM A$(1)*40 [ENTER] then A$(1)="XXXXX" [ENTER], and A$(1)=A$(1)+A$(1) [ENTER].
Now press the ◀ cursor, and your last statement is redisplayed.
Key [ENTER], and it will be executed again. Once more, ◀ cursor and [ENTER]; and A$(1) consists of exactly 40 Xs.

MINDBOGGLE CORNER

No entries, of course, for the January competition. Nor were there any for the Golf competition. The February deadline is not yet, but I shall be surprised if there are any entries for this either. Anyway, here is the answer to the January puzzle.

Take each character in the first statement   Z>)THO!:1>TJO!:1*   and subtract 1 from the ASCII code of each character. The resulting codes will give: Y=(SGN 90=SIN 90)   And the answer therefore is 1. What could be simpler! The reference to 121 and 125? The code for SIN is 241 125, and 241 121 is the code for SGN.

This months MINDBOGGLER is a genuine one; I do not know the answer. On page 8 was a suggested subroutine for varying DIMensioning according to requirements. Now if you use this in a program, and save the resultant data after running the program, how then do you load the data again? You have to DIMension space before INPUT# N$(*): - but the details of what space to DIMension are in the data which you cannot yet load! Suggestions, if any, by April 15th, please. "Write down the details" is not an acceptable answer. Usual prize is offered for the best solution.

GETTING STARTED ON THE PC-2 book review

This book is written for mentally defective illiterates. I found it useful. It is not intended as a substitute for the instruction manual, but as a supplement to it. You are taken patiently step by step through the various commands and statements, with ample illustration. Where a term is hard to understand, such as "syntax" or "array" it is explained with nauseating little analogies and anecdotes. Helpful to the beginner are examples of common MISTAKES in the use of each command or statement - though the expression "OOPS! WHAT WENT WRONG!" becomes cumulatively more and more irritating.

But while the transatlantic use of the English language, on this childish level, can be annoying, it is at least comprehensiblé, and many things are explained much more fully, and much more clearly than in the manual for the PC 1500. In particular, it makes sense of the various forms of PRINT# and INPUT#, with and without FILENAME, in a way that I can at last understand and remember. Most of all I like the section at the end of the book, where a few programs are given, with a simple explanation of the effect of each separate statement, in parallel. Anyone meeting BASIC for the first time will find this most educative.

Very few of the assertions in the book are actually completely wrong. The authors say that when poking a number into a memory location, both number and location being described by expressions, such as poking A*X into location B+Y, then the use of brackets is essential. In fact the example they give:

```
POKE (1000+A,100+B)
```

will only produce ERROR 18. The alternative version:

```
POKE 1000+A, 100+B
```

without the brackets, gives no ERROR. The limit of character size is given as CSIZE 9, and the limits of Reverse Line-Feed are also clearly illustrated. Those of you who followed this newsletter last year will know better. Indeed, no really useful example of PEEK or POKE in action is given, despite the space devoted to these statements. I get the impression that none of the 3 authors has ever actually experimented with the machine itself, but that they have relied on their knowledge of computers in general, and on the intelligent interpretation of written material - itself no mean feat, in the case of English instructions written in Japan. However I do take exception to their assertion that POKE is not dangerous. In my own early experiments I must have "crashed" at least ten thousand times, and certainly lost programs which were tedious to replace. Either the authors bear charmed lives - or as I have suggested, experimented only on paper. Nevertheless the book at least introduces the reader to PEEK and POKE and CALL, which is more than the PC 1500 instruction book does.

On the credit side, I very much like the clear exposition of all the commands and statements in the Appendix. I like too the way the ERROR codes are dealt with, including an example of how each can occur. The PC 1500 instruction manual often describes these so tersely that one is none the wiser. I have reviewed the faults of this book at greater length than the virtues: but in fact the virtues far outweigh the faults, particularly for the absolute beginner. I know that a number of our readers have found it most helpful, and if it had been available to me when first I started with the PC 1500, I should have been very grateful indeed.

GETTING STARTED ON THE PC-2 is published by TANDY at £8.95

MARKETPLACE

TANDY are no longer manufacturing the PC-2.
They will however continue to support machines already distributed.

The price of the TANDY interfaces has been reduced to Ł69.95, which compares very favorably with the price of the CE 158. However it must be noted that the TANDY interfaces are RS232 only .

Also reduced in price is the TANDY range of software for the PC-2. Prices are now about Ł5.95. These prices are subject to change, and may vary locally.

An EPROM programmer is available in USA. This of course is mainly of interest to large business users. If any reader is seriously interested I can supply details, and may be able to import the machine and the progammable EPROMS. I believe that they may in fact not be true EPROMS, but battery-supported RAM modules.

I hear from a reliable source that the delivery situation for the TRAMSOFT TOOLKIT 2 from WALTER SPIEDEL has improved dramatically. A reader who recently purchased one received it within a fortnight of sending for it. The price is DM.371, which is just under Ł100 at the present rate of exchange.

CAPSET, which was reviewed in volume 1 (page 100), is now being marketed by MINIMICRO SOFTWARE, and distributed by ELKAN. Some readers have found the program useful for designing layouts: I understand that a few slight improvements have been made.

SUPERTEXT updates are enclosed with this issue to those who possess this program. The cassette, updated, is still available only to subscribers at Ł7.95. (8K is essential). A more sophisticated version, requiring 16K extension, and specially adapted for use with a printer (such as my JUKI) will be available shortly at Ł13.95. As soon as it is finalised, current owners will be able to obtain the revised program, on cassette, on payment of the difference. (Much of this edition was prepared with SUPERTEXT).

A few more details about the JUKI 6100 PRINTER. This cost Ł399 from CORTEX COMPUTERS, (6-10 Great Portland Street). The CE 158 is essential. I also needed the EA158C parallel lead. SHARP (UK) who theoretically stock this, were as usual unhelpful: ELKAN were able to obtain one for me; it took about a fortnight. Alternatively, the JUKI is available with RS232 interface, at a surcharge of Ł65, and thus one would be able to use one of the cheaper TANDY interfaces.

*

STATUS 1500 (volume one) is now available, bound and indexed. I need hardly say how much valuable information it contains.

Prices - inclusive of postage - are:

UK: Ł8.50
overseas (airmail) Ł13.50
overseas (surface mail) Ł10.50
Allow about 14 days for delivery in UK.

*