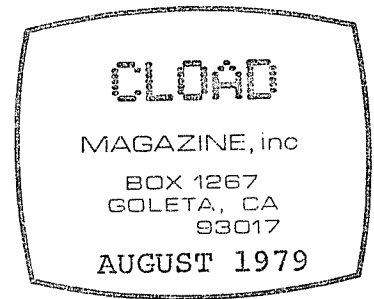


The CLOAD to disk connection

Many of our subscribers have written or called us describing various problems that they've encountered while loading our programs. Lately, a common problem seems to be the fact that level II Disk BASIC does not always seem compatible with level II (non disk) BASIC. This problem is particularly evident when the program in question makes use of PEEKS and POKES, or machine language subroutines. The only realistic solution is to step back to the 16K, level II non - disk configuration for these programs.



The heart of the problem runs along the same general lines as the Biblical story of the Tower of Babel. Systems level programmers have developed the concept of "upward compatibility" which allows a language to grow and become more powerful while allowing programs written in the previous version to run without changes. Unfortunately there exists a Force in the local universe which prevents a perfect implementation of this concept. I personally accept it as quiet confirmation that the universe is structured and orderly that mere mortals cannot perform this feat, else humanity would build a tower, of this thought-stuff we call "software", which would penetrate the ozonosphere and thereby violate innumerable zoning ordinances.

So our ivory towers are still constructed of real and substantial white bricks, and our software is continually being rewritten from scratch, to its infinite benefit.

A short remark on REMark statements: use them in your programs! BASIC is a powerful language, but it has several shortcomings when the time comes to repair or change your code. The REMark can alleviate quite a few of these shortcomings. My pet peeve with BASIC is the fact that lines of code can only be referenced by numbers. Thus we have GOSUB 2010 instead of GOSUB UPDATE. We can fake it with GOSUB 2010 : REM UPDATE TIME COUNT. Another confuser is GOTO 560. The GOTO statement actually has two operands - where it is, and where it's going. Say GOSUB 560 : REM CLOSE LOOP, REENTER AT VARIABLE TEST. A reasonable standard for REMarks should be as follows:

- 1) At the beginning of each program - some sort of title
- 2) As each variable is initialized - what that variable represents.
- 3) At each GOSUB that hasn't been called in the last few lines - what the subroutine would be named if that were possible.
- 4) At the subroutine itself - input variables, output variables, scratch variables, action of subroutine.
- 5) At each GOTO statement - why it's there and where it's going. The GOTO statement should get comments above all others.

The main dissertation this month is the use of arrays in mathematical modeling. For an example, we'll use the old maximize-the-income problem.

Let's first make the simplifying assumption that we are in a business where we buy wholesale and sell retail. The traditional thrust of marketing in such operations is to adjust our retail price level to make our goods attractive compared to the competition. Let's look at a typical item: The high price limit is the price at which sales drop off. The low price limit is the price at which too much money is lost selling the item. Within these two limits, our income is the number of sales times the unit profit.

Looking at these two variables - sales volume and unit profit - we see that they are both largely determined by the price we set. It would be nice to have a method of taking sales volume as a function of price and multiply it by unit profit as a function of price and arrive at net income as a function of price. And after having spent this much time leading you up to the problem, would I shirk my implied obligation to follow through? It's tempting...

```

10 REM EXAMPLE OF SALES VOLUME GUESSER
20 DIM A(20) : REM A ARRAY IS SALES VOLUME VECTOR
30 FOR I=1 TO 20 : REM 20 PRICE LEVELS CHECKED
40 P=10+I : REM STARTING AT $11 (OUR COST IS $15)
50 PRINT"HOW MANY WIDGETS PER MONTH WOULD WE SELL AT $";P;
60 INPUT V : REM SALES VOLUME PER MONTH
70 A(I)=V : REM TUCK IT AWAY
80 NEXT I
90 CLS : INPUT"SET UP A BLANK TAPE FOR RECORDING AND HIT ENTER";X$
100 REM HOLD AT LINE 90 TILL TAPE IS SET UP
110 A$="WIDGETSALESVOLUME$11TO$30" : L=20 : REM FILE LABEL, LENGTH
120 PRINT#-1,A$,L : REM PASTE A HEADER ON THIS HERE FILE
130 FOR I=1 TO 20 : PRINT#-1,A(I) : NEXT I
140 REM ARRAY IS NOW SAVED ON TAPE

```

What is this? guessing? Why not guess at the final price and be done with it? Well, I'll admit that this is not a method that has the aura of scientific precision even if it does use a thinkum machine, but the general idea is to separate out the various factors that lead to a decision, so that influencing factors (such as the competition's price) don't assume too much weight. A lot of hobby computer manufacturers have gone bankrupt recently as a result of underpricing the competition, who were themselves underpriced.

Therefore, let us barge on. We have an array of monthly sales volume as a function of price, let's make another array of unit profit as a function of the same price range. The program looks very similar:

```

150 REM NEXT WE GO FOR UNIT PROFIT
160 DIM B(20) : REM B ARRAY IS UNIT PROFIT VECTOR
170 FOR I=1 TO 20 : REM SAME 20 PRICE LEVELS AS ABOVE
180 P=10+I : REM SAME STARTING PRICE
190 PRINT "WHAT IS OUR NET PROFIT PER WIDGET AT $";P;
200 INPUT M : REM MONEY MADE PER UNIT SOLD
210 B(I)=M : REM STUFF IT IN B ARRAY
220 NEXT I
230 CLS : INPUT"SET UP TAPE FOR RECORDING, THEN HIT ENTER";X$
240 REM X$ ISN'T USED EXCEPT TO HOLD UP COMPUTER
250 A$="WIDGETSALESPROFIT$11TO$30" : L=20 : REM LABEL,LENGTH
260 PRINT#-1,A$,L : REM HEADER AT THE HEAD
270 FOR I=1 TO 20 : PRINT#-1,B(I) : NEXT I
280 REM B ARRAY NOW MAGNETIZED

```

Sharp observers might note that there is a lot of program duplication going on. While the program could be made a lot shorter by merging it with the one above, think of how it works at RUN time. The (hopefully) human operator will find it easier to handle one subject at a time. Remember: Human - master; Computer - slave. Say it ten times.

While we're observing, let's point out the things that should be considered while filling out these arrays.

Sales volume - if the price is set below cost, is the volume limitation going to be availability? The number of customers already in the store? The rate at which the product can be delivered? Or is it a relatively constant demand item, regardless of price (like gasoline). As the price increases past the competition's, will the customers think there's a quality difference? Would this affect sales volume?

Unit profit - obviously selling below wholesale involves a loss - how much? Consider the current tax margin (losses are deductible). Consider the proportional cost of labor involved in selling the item. As the sales price moves above "break-even", how much of the gross profit is kept (again, considering tax margin)? Does anyone else share the profits of this particular item? How much?

O.K. - we've got some estimates on the market. Since net income = sales volume * net profit per item, our income picture is revealed by $A(I)*B(I)$. Stand back! We're going to form the dot product of two vectors!

```

290 REM DOT PRODUCT OF A AND B
300 FOR I=1 TO 20 : REM EACH PRICE LEVEL
310 C(I)=A(I)*B(I) : REM C(I) IS NET INCOME AT THE I'TH PRICE LVL
320 NEXT I : REM THAT'S ALL THERE IS TO IT, FOLKS
330 CLS : REM LET'S DISPLAY NET INCOME AT EACH PRICE LEVEL
340 PRINT"SALES PRICE, THEN NET INCOME"
350 FOR I=1 TO 20 :REM 20 PRICE-INCOME PAIRS
360 PRINT I+10, C(I), : REM REMEMBER A(I) PRICE OFFSET
370 NEXT I
380 PRINT : PRINT"SET UP FOR RECORDING, THEN PRESS ENTER";X$
390 A$="WIDGETNETINCOME$21TO$30" : L=20 : REM LABEL (AGAIN)
400 PRINT#-1,A$,L : REM I SHOULD HAVE WRITTEN A SUBROUTINE
410 FOR I=1 TO 20 : PRINT#-1,C(I) : NEXT I
420 REM C PUT ON TAPE WHILE MASTER PERUSES GARBAGE ON SCREEN

```

The intelligent question to ask at this point is: so what? Well, what we have here is information to base business decisions on. It can only be as accurate as the input information (the Garbage In Garbage Out principle). Apparently, the proper thing to do is to price this item to achieve maximum net income according to our estimate. This may or may not be the right course to take. It might not cost too much to carry this item as a loss leader - supermarkets have priced coffee this way for years. The current price might be way off the mark, but for other reasons we may choose to let it stand. This is proper, too. Human - master; Computer - slave. The important thing is that we now have information on the cost of doing so.

Let's take an overall look at what we've just done - remember that this is supposedly just an example of using arrays in a mathematical model. A real program which accomplishes what we're illustrating would have to take all the other things we sell into consideration, as well as quantity discounts, stock aging and the like.

We proposed a problem that had several interacting variables, and used one of them - price - as a base to work with (note that it was the variable over which we had the most control). We then made the presumption that the other variables - volume and profit - were directly associated with this base variable (not true but tolerably close). We then filled out a one dimensional array for these associations (note also that the most accurate data available in this case was an experienced guess - if there's better data, use it). We then had all we needed to form a third array, which showed us a relationship much more clearly than our previous hunches.

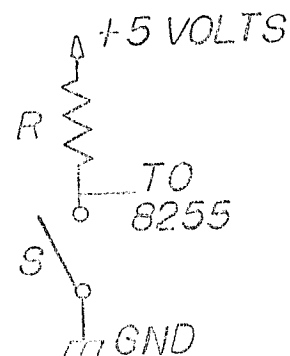
Last month it seems that our master tape got a glitch in the program PSYCHO. If your copy does not work, the following lines should be checked. The troublemaker is line 1180 most of the time.

```

1100 DATA17,1,60,1,0,4,54,128,237,176,221,33,8,114,62
1110 DATA1,33,8,114,6,4,119,35,16,252,221,126,0,221,134
1120 DATA2,221,119,0,221,126,1,221,134,3,221,119,1,221,126
1130 DATA4,221,190,0,48,22,221,126,2,47,221,203,6,78,32
1140 DATA1,60,221,119,2,203,39,221,134,0,221,119,0,221,126
1150 DATA0,230,128,32,227,221,126,5,221,190,1,48,20,221,126
1160 DATA3,47,221,203,6,70,32,1,60,221,119,3,221,134,1
1170 DATA221,119,1,221,126,1,230,128,32,229,221,86,0,221,94
1180 DATA1,205,216,113,221,203,6,86,32,20,62,127,146,87,205
1190 DATA216,113,62,47,147,95,205,216,113,221,86,0,205,216,113
1200 DATA58,186,114,183,40,8,71,62,255,61,32,253,16,249,58

```

Last month we ran out of time and couldn't continue the hardware talk. This month let's set up a switch as a computer input. As you can see by the diagram, this is a rough circuit. It involves a switch, a resistor of about a thousand ohms (500 ohms is a tad low, and 10,000 ohms is a tad high, but anything in between is OK - 2200 ohms is a popular value). If the switch is closed (allows electrons through) the junction between the resistor R and the switch S will go to a low state (less than .2 volts - a logical 0 output). Conversely, if the switch is open (no electrons allowed through), the junction will go high (more than 2.8 volts). This junction is connected to an input pin of the 8255 I/O chip.



NEXT MONTH!

Ralph

Box 1267
Goleta, CA 93017
(805) 964-2761
MasterCharge/VISA
welcome

===== C L O A D M A G A Z I N E =====

SUBSCRIPTIONS \$36.00 IN NORTH AMERICA
 \$38.00 OVERSEAS SURFACE
 \$48.00 OVERSEAS AIRMAIL

BEST OF CLOAD, VOLUME ONE
 \$10.00 (\$15.00 OVERSEAS)
 CA RESIDENTS ADD 6% TAX

SINGLE ISSUES \$3.50 EACH PREPAID + CA TX

T-SHIRTS - CLYDE CLOAD IN BLACK ON FRONT
 S, M, L, XL - BLUE OR YELLOW
 \$7.00 EACH PREPAID + CA TX

LEVEL I PROGRAMS ARE WRITTEN FOR 4K
LEVEL II PROGRAMS ARE WRITTEN FOR 16K
SUBSCRIPTION STARTS WITH CURRENT ISSUE

