The Amateur Computerist

April1, 1989 Volume 2 No. 2

WHY LEARN TO PROGRAM?

Each issue we would like to have a debate or discussion over some issue facing computer users. We invite opposing viewpoints on articles presented here and hope to stimulate debate, as development only comes through exchange and debate. Following is a brief excerpt from a discussion stimulated by the "Why Learn Programming?" article, Vol 1, No 1.

OPPOSING VIEWPOINT

Although I learned programming when I first started to use the computer via typing in programs from magazines, I don't think it is necessary to learn to program to use a computer.

I feel in order to operate any word processor, for example, knowing BASIC isn't a requirement. It isn't a requirement but it is a help.

It isn't a requirement. I feel good reading skills and good interpretation of what you read is more important than knowing BASIC or PASCAL or any other programming language. The manuals are usually laid out pretty good. They're good enough to get you to do what the program is intended to do. As far as WordPerfect, for example, as far as you getting to the language the program is written in, you might as well forget it. Its compiled so you can't get to the language anyway.

You should spend your valuable time learning the features of the program – you'd be better off. The problems people usually have – they wouldn't have if they read the manual thoroughly or they practiced enough. You have to practice with it to get good at it.

If you have trouble with a word processor, its because you don't know the features and functions. Once you know them for one word processor, its easier to learn them on a different one.

Steve A.

DISCUSSION

Mike: You're saying that programming isn't a requirement, but it is a help? What you're saying isn't against learning programming, but you're just saying you don't need it?

Steve: The problems people usually have they wouldn't have if they read the manual thoroughly or they practiced enough. You have to practice with it to get good at it. If you have trouble with a word processor, its because you don't know the features and functions. Once you know them for one word processor, its easier to learn them on a different one.

Mike: Do you feel that if you didn't have BASIC you wouldn't be as advanced as you are now?

Steve: It's hard to say.

Mike: Do you ever fall back on it?

Steve: No, but I found I had a lot of fun with BASIC. I made a program for lottery numbers, and it was easy to add onto it, add graphics, etc.

Mike: Do you think BATCH programming is necessary for your computer?

TABLE OF CONTENTS

Steve: I don't think anyone could maintain their files without making BATCH files. That's the whole secret of using your computer. When you turn on your computer, it goes to that AUTOEXEC.BAT file. You can change your procedure by changing the AUTOEXEC.BAT file. I have a hard disk menu program that's real nice. In order to get it to run effectively, you have to configure your autoexec file so it will come back to that hard disk menu.

Ronda: Where's the disagreement?

Norman: It all depends on what you want to do. If you want to do programming, you have to go into programming.

Steve: If you want to use your computer for word processing, spreadsheet, data base, then you don't have to go into programming.

Bill: Like if you want to use the radio on your car, you don't have to learn how it works.

Steve: I'm glad I do know some programming. I'm going to try to learn more. One thing, it’s a challenge. My boy won't come within 20 feet of a computer because it’s a challenge. If I can ever break that down.

Mike: What programming does, it makes people less intimidated by the computer. I can do what I want with it.

Steve: Not only that, you can go one day and start from scratch. If you know what you want to do, you end up with one sheet, and it works! You've done something. Add colors, etc. Pretty soon you've got a nice full blown program. You got the program to do what you want it to do. When I didn't know how to do something, I'd go to another program that did it, and I'd copy the structure and low and behold, it would work!

I used to be good at typing these programs out of *RUN Magazine*. You'd get a syntax error. You'd have to go back to that line. If you're lucky enough you get the program to run and figure it out.

Mike: It might be in the magazine that its wrong. If you know programming you can figure out the error.

Steve: The computer can be intimidating. People think its going to make them look like a fool.

Ronda: The computer represents a stupendous technological advance. Computer technology is only in its infancy. We don't yet realize what its capabilities are. It's been a 400 year effort to develop the personal computer and now it's not only here, but it's also in people's homes.

Mike: Are you, Steve, saying programs are already written and so you don't need to learn to write programs?

Steve: I think there should be programming classes but you don't have to learn to program to use a computer.

Norman: Are you, Ronda, saying that it should be mandatory to learn BASIC or programming?

Ronda: I'm not saying make it mandatory, but I am saying that there needs to be strong support and encouragement for computer users to learn to program.(* see note)

Norman: First of all, I'd like to know the reason why they don't want people to program. Maybe the company just wants to teach you what they want you to know. When we wanted advanced classes, we were stepping beyond the limits of what they wanted us to have. When we finished Computer Literacy I, we wanted more. Even though I didn't have a computer I was hungry for more knowledge about computers.

Steve: When I go to a shopping center, I type in a program "What is your age? Put in your age. "Then I type in "You look pretty good for your age." and I put it in a loop. Its nice to be able to do that.

*** ADDITIONAL COMMENT:**

Computer users are now told "You don't have to learn to program to use a computer." Or they are told to read their manuals if they run into trouble with a program.

But the problems users have with computers is most often not because they don't know how to read directions, or because they haven't studied their manuals enough. The problem is most often that users don't understand how computer technology functions and thus have unrealistic or wrong expectations of the computer. But those users who know some programming, know how the computer functions and so have the basis to troubleshoot any unexpected or difficult situations. (In fact they know to expect difficulties, not to expect using a computer to be easy.) Software writers can only anticipate some of the problems with computers. They can't anticipate all the uses or problems that may occur. Hence there will be bugs, not ease, in using computers at our current state of technological development. And a broad knowledge of programming among computer users will make it possible for users to customize the software for their own needs. It's as if someone said "You don't need to learn to read. Someone can read to you," when the printing press was first invented.

Ronda

LETTERS

Noting Ssgt. Campbell's letter in *The Amateur Computerist* (vol 2 no 1, ED.), I can provide a few tips that may help to locate some (unfortunately, not all) of the components he needs to complete his MC68000. There are a (very) few dealers catering to the hobbyist or hacker market who stock ALS parts. Perhaps the best is Active Electronics, P.O. Box 9100, Westborough, Ma 01581. Their current catalog lists all the ALS items that Campbell is seeking. Other suppliers that I know of are Jamesco Electronics, 1355 Shoreway Road, Belmont, CA 94002, and BG Micro, P.O. Box 280298, Dallas, TX 75228. Most of the mainline distributors carry ALS, but may have large minimum order requirements.

According to my out-of-date (1982) copy of IC Master, Texas Instruments TMI2149-3 (35-nsec) or TMI2149-4 (45 nsec) should be acceptable substitutes for the NEC 2149-2. Another substitute is the Fujitsu MBM2149-45. You could try Microprocessors Unlimited, 24000 S. Peoria Ave., Beggs, OK 74421. They specialize in memory chips and offer superlative service.

Jamesco stocks 68000 and 68010 CPU's.

The remaining parts that Campbell mentions pose a problem. Neither my IC Master nor any data books that I have at hand list either the 74LS172 or the 9229-B. It does list the plain vanilla TTL 47172 16-bit register file. This may be a workable substitute, if you can finds one. I don't know what the 9229-B is. Some 9000-series chips are the same as the corresponding 8000-series numbers. However, I don't find any listing for an 8229, either.

Usually the most direct source for hard-to-get parts for a kit is the kit manufacturer or distributor. This, of course, is out if the system becomes an "orphan" (i.e., the supplier goes out of business). This may be the case with Computer Express. In this case it's up to your own ingenuity or design expertise to work around the problem and try to salvage your investment. I would fault *Radio-Electronics* for being lackadaisical at best in their promotion of the MC68000. If you wrote again to ask if any of their readers could help, they ought to at least to publish your letter. The PT68K seems like a good system that is still being supported and Peter Stark has always been a superb instructor. It does use a 1772 floppy disk controller, which seems to be an elusive item. I wrote to Stark to inquire if this was the correct part number, but received no reply. This sort of turned me off, though I may yet construct the system as a learning experience and a stepping stone to a 68020 system.

Please keep *The Amateur Computerist* coming. It's not really up to speed yet, I'd say, but is on the way (to compound metaphors!). Readers' letters should be the mainstay. it would be helpful, though, if you'd publish complete addresses with the letters, to facilitate communication.

Very truly yours,
Norman F. Stanley
Rockland, Maine

(Editor's note: Starting with the next issue, full addresses will accompany letters unless writers request their addresses be withheld.)

I hope you had a Merry Christmas and the New Year finds you in good spirits.

I am very pleased that you liked my suggestions, and that you will publish my letter in this months' newsletter.

When you asked for more suggestions, I couldn't think of any until I saw a book printed by the media institute, entitled "Using the New technologies." Then I saw the program the "Computer Chronicles." I put the two together and figured that the *Amateur Computerist* could use cable to inform the country what we're about. Our format could be to have people (newsletter subscribers) write in about a problem they are having and if we get enough of the same problem, put it on the air. Or use a modem to a television screen and work that way. The main part of the program should be teaching (or instructing) Logic and computer programming, MS-Dos etc. There isn't a program on the air at this moment but we could be the first. As I said, this is only an idea, but if someone was to write a good proposal to the National Science Foundation or to any of the computer companies, they would give us partial funding.

Now I have to ask about courseware, computer software for instructing educational courses. Do you have any good courseware on the college level? If not this is another area we should investigate.

I have some more ideas but I'll write them in my next letter to you.

Very truly,
Gary Chestnut
Prairie View, TX.

I just completed rereading the last copy of *The Amateur Computerist.* I believe that your efforts are great. I am impressed with Michael's research and writing abilities and I share his views on free speech (Ed. note: Vol 2, no.1). I don't necessarily agree that the failure to develop the personnel computer is related to the lack of freedom of speech. As I have stated early I believe that the incentives for not proceeding with personnel computer development were certainly economic.

I know that you are aware of the factor that historically that much of the technological advancement made in the past resulted from the work of mavericks who are outside the mainstream. Such people were often ostracized for their efforts and they prevailed in spite of many obstacles. If you want to call the discouragement by the power groups the lack of free speech I guess it is one way of looking at the problem. As I stated in my earlier letters I believe that what happened with the personnel computer is typical of the development of all technologies, where people will not go forward with something new when they have a profitable alternate. Also suppression of innovations by the existing power structure is also based upon ego (particularly in the scientific community.)

I was a little disappointed in not seeing any comments regarding my letter mentioning "Big Brother" and its potential impact on society. Certainly this is going to be one of the most negative aspects of the computer revolution. I believe that this problem should be addressed more and potential impact documented, in any studies on the effects of the personal computer. I don't know how we can stop this trend, but it is certainly deserving of much study and further comment.

Yours truly
David Pollack
Sun City, Az

I am interested in the technical side of computers and interfacing computers (Commodore 64, IBM compatible), to other equipment.

I am also interested in an article on what is required to interface a CD disk player to be able to use the CD Data Bases that are coming on the market, and also connectable EPROM programming.

Most of the computer magazines on the market only feature software.

According to the article in *Radio Electronics*, your Newsletter should offer some assistance in this regard.

Please send subscription rates as I would like to subscribe to the *Amateur Computerist Newsletter*.

Thank you and I look forward to learning new ideas on computers from your publications.

Yours truly,
D. McKague
Ontario Canada

I am writing to you regarding your newsletter *The Amateur Computerist*, which I read about in *Factsheet Five* #28. I am a Health and Safety Officer for my local and we have been putting together a Technical Change Committee. I have been asked to serve on the committee, but I know very little about computers (I am a Boiler Engineer).

Your newsletter sounds like just the sort of thing I need. Unfortunately there was no mention of foreign subscriber price. Could you provide me with this information as quickly as possible? Thank you for your help.

Sincerely yours
Randall Barnhart
Ontario Canada

Some people are worried that if the workers win a six hour day, thirty hour work week at forty hours' pay, their employers will put in as many computers, labor-saving machines, and robots as it possibly can. Then, these people reason, instead of saving jobs, the six hour day might even lead to more workers out of work. A look back in history may be helpful in addressing this concern.

Around 1850, the working class in England finally won its one half century struggle for the Ten Hour Day. The English employers did then in fact try to make up for the lost man hours. The English manufacturers put into place, sparing no cost, the most advanced machinery of the day. And what was the result? These manufacturers were able, because of the newest machinery, to produce more and better products than they were producing before the shortening of the work day. Because these products were produced in fewer man-hours, they could be sold at lower prices. Demand for these products increased. Instead of the anticipated loss of jobs because of such wide spread use of new machinery, all sectors of English society, including the working class, experienced a fifteen year upsurge. In particular, there was an improvement in the number of jobs available and in the living standard of the great majority of the English workers.

This lesson from history is important for today. It suggests that the American Labor Movement should put on its agenda a struggle for the six hour day at eight hours pay. By so doing, it will not only be fighting for jobs at a necessary minimum wage for all workers, but it will start the ball rolling to put on all American manufacturers the only pressure they respond to, the threat of lower profits. Until manufacturers are faced with the loss of profit rates that the six hour day at eight hours pay means for them, they will not have the necessary economic incentive to put into place expensive, state of the art computer-aided machinery and robots. The fact is that manufacturers always resist the high investment necessary for new machinery and technology as long as they can stretch the working day or lower wages. That is what is happening today and that is why there is a loss of auto jobs, for example. And as long as auto manufactures continue this short sighted intensification of work under the guise of meeting competition, jobs will continue to be lost. The workers who win the shorter work day at full time pay are the ones who will be best protecting their jobs and the jobs of future workers.

My conclusion is, "Yes, a victory for the Six Hour Work Day Movement will be met by the introduction of labor-saving computer-aided machinery, robots and computers." But in fact that will be in the interest of workers because prices, for example on cars, will come down, more cars will be sold, and the number of people working to produce cars will in all likelihood increase, not decrease. It is only the introduction of new machinery that can have any effect in slowing or stopping the current downward trend of auto jobs. Therefore the Six Hour Work Day at Eight Hours Pay is a proper demand for auto workers.

Jay Hauben
Dearborn, Mi.

TRY THIS

"Message" for IBM in BASIC

by Michael Hauben

(Editor's note: This program lets you type in a birthday or holiday greeting. To exit press Control-C.)

```
10 REM MESSAGE
20 REM COPYRIGHT 1985 M. HAUBEN
30 SCREEN 2: KEY OFF
40 CLS
50 LINE INPUT "Please type in a message (up to 50 characters): "; A$
60 CLS
70 A$ = " " + A$ + " "
80 LINE(190,80) - (200 + 8 * LEN(A$), 104), 1, B
90 FOR I = 1 TO LEN(A$)
100 LOCATE 12, 25, 0
110 PRINT MID$(A$, I); LEFT$(A$, I)
120 FOR A = 1 TO 200: NEXT
130 NEXT I
140 GOTO 90
```

"SE Q" for IBM (or Commodore)

by John Primm

(Editor's note: This BASIC program gives the next number in a series.)

Some examples users can try when running the program are:

Numbers to type in	Next number
2 4 6 8	10
2 4 8 14	22
1 2 6 24	120
3 6 11 18	27

(This program was written based on an algorithm in *Scientific American*, March 1986, pg 14-21.)

```
5 CLS: REM Commodore will use PRINT"{CLR}"
10 INPUT "Amount of numbers to be entered (minimum of 4)"; N
30 FOR I=1 TO N: REM loop for array
35 PRINT "Enter number ";I;
40 INPUT A(I)
50 NEXT I
60 FOR K=-5 TO 5 STEP 1:REM loop to set variable
70 GOSUB 200: REM try 2nd level diff. form
80 IF M=1 THEN M=0: S=1: A$=INKEY$
90 NEXT K
100 IF S=1 THEN END
110 PRINT "HELP": END
200 REM 2nd level diff. form.
220 FOR I=1 TO N-1
230 B(I)=A(I+1)-(K*A(I))
250 NEXT I
260 GOSUB 400: REM try 3rd level diff. form.
270 IF M=1 THEN GOSUB 900: RETURN: REM print 1st level diff.
280 GOSUB 300: REM try 2nd level quotient form.
290 RETURN
300 REM 2nd level quotient form.
310 T=1
320 FOR I=1 TO N-1
325 IF A(I)=0 THEN 380
330 B(I)=(A(I+1)-K)/A(I)
350 NEXT I
360 GOSUB 400: REM try 3rd level diff. form.
370 IF M=1 THEN GOSUB 1000:REM print 1st level prod.
380 RETURN
400 REM 3rd level diff. form.
410 FOR I=1 TO N-2
420 C(I)=B(I+1)-B(I)
430 NEXT I
440 T=1
450 FOR I=1 TO N-3
460 IF C(I)=C(I+1) THEN T=T+1
470 NEXT I
480 IF T=I THEN C=C(I): KK=K: PRINT "3rd level is developed as C("; N-1; ") = B("; N-1; ")-B("; N-2;") and is": GOSUB 600: REM print 3rd level
490 IF M=1 THEN GOSUB 700: RETURN: REM print 2nd level diff.
493 GOSUB 500: REM try 3rd level quotient form.
497 RETURN
500 REM 3rd level quotient form.
510 FOR I=1 TO N-2
515 IF B(I)=0 THEN 595
520 C(I)=B(I+1)/B(I)
530 NEXT I
540 T=1
```

```
550 FOR I=1 TO N-3
560 IF C(I)=C(I+1) THEN T=T+1
570 NEXT I
580 IF T=I THEN C=C(I): KK=K: PRINT "3rd level is developed as C("; N-1; ") = B("; N-1; ") / B("; N-2; ") and is":GOSUB 600: REM 3rd level
590 IF M=1 THEN GOSUB 800: REM print 2nd level prod.
595 RETURN
600 REM 3rd level printout
610 M=1
620 T=5
630 FOR I=1 TO N-1
633 PRINT TAB(T); C;
637 T=T+4
640 NEXT I
650 PRINT: PRINT
660 RETURN
700 REM 2nd level diff. printout
705 PRINT "2nd level is developed as ";
707 PRINT "B("; N; ") = B("; N-1; ") + C("; N-1; ") and is "
710 T=3
720 FOR I=1 TO N-1
730 PRINT TAB(T); B(I);
740 T=T+4
750 NEXT I
760 B(N)=B(I-1)+C
770 PRINT TAB(T); B(N)
780 PRINT
790 RETURN
800 REM 2nd level prod. printout
805 PRINT"2nd level is developed as ";
807 PRINT"B("; N; ") = B("; N-1; ") * C("; N-1; ") and is"
810 T=3
820 FOR I=1 TO N-1
830 PRINT TAB(T); B(I);
840 T=T+4
850 NEXT I
860 B(N)=B(I-1)*C
865 PRINT TAB(T); B(N)
870 PRINT
880 RETURN
900 REM 1st level printout
905 PRINT"1st level is developed as ";
907 PRINT "A("; N+1; ") = ("; KK; " * A("; N; ")) + B("; N; ") and is"
910 T=1
920 FOR I=1 TO N
930 PRINT TAB(T); A(I);
940 T=T+4
950 NEXT I
960 PRINT TAB(T); B(N)+(KK*A(I-1))
970 PRINT
980 RETURN
1000 REM 1st level prod. printout
1005 PRINT"1st level is developed as ";
1007 PRINT"A("; N+1; ") = A("; N; ") * B("; N; ") + "; KK; " and is"
1010 T=1
1020 FOR I=1 TO N
1030 PRINT TAB(T); A(I);
1040 T=T+4
1050 NEXT I
1060 PRINT TAB(T); B(N)*A(I-1)+KK
1070 PRINT
1080 RETURN
```

AS I WAS SAYING...

by Floyd Hoke-Miller

JOBS: Hours and Sense Not Dollars and Cents

Jobs! What a word of terror it has become in the hearts of those depending upon the workplace for subsistence. What an enigma it is as presented by the government, the media as a whole and even by the labor "leaders" in part. "Jobs, Jobs, Jobs", "Jobs Lost", "Job Security" they shout. But the ultimate job security is chattel slavery: a guaranteed job for life and for your children after you — the only rub being no pay, no freedom, no life. At the wage-slave level, this is a tragedy of deep felt proportions by all those who had hoped for the correction to come from the realm of Organized Labor.

Jobs. A word so bandied about but out of logical presentation. You hear it as a curse: "Your job may be gone soon." You are told that is the effect of competition, of new technology, of cheaper labor elsewhere, of.... But you hear nothing of the simple and logical cure: Shorter Hours of work for all. Share the work, not the misery.

Jobs. Much used but little understood, it was the main theme of both major parties in the recently played out circus, "the World's Greatest Show", the U.S. Presidential Election campaign. Both rings vied with each other for the voter's attention with the same script, differing for the record only in the number of jobs promised if elected. The wise among us branded this as a farce, this promise of Jobs, Jobs, Jobs. Now the race is over and they have been proven right. GM has announced more jobs to go.

Jobs. Seldom before in the history of politics was there more evasion, gobbledegook so to speak, more chaos purposely spewed for mass consumption and confusion as about jobs. REASON, a democratic device of weighing the plus and minus factors, of sifting facts in search of truth was never employed in the campaign only perverted. Not one, not a single performer presented the cause, the truth that new technology could be a boom to workers, making possible the shorter work day and thus longer time for study, thought, family, leisure, rest, sport, etc. Instead, their trick was the Big Threat: Be obedient, well trained workers, taking less for doing more, forget your union, work longer hours, be a team player with management or – your job will go elsewhere.

Jobs. Behind the threat of their loss is the concentration of power and pelf in fewer and fewer people and the subservience of the press and all of academia to them. Together they make every effort to hide the fact that Hi-Tech can either be a boom for workers by allowing shorter hours or a bust by allowing the threat of elimination of the human factor (which is never really possible) from the creation or recreation of goods and services. The Big Show politicians of both major parties represent this minority control of the Cybernetic Revolution. This revolution, the coming of the use of computers and robots combined to do much of the needed work, could be the culminating end of the Industrial Revolution in our transition from the Dark Ages to the Age of Enlightenment. But not if used to stretch the work day and further enslave workers. Check the parallel in history of the abuses of the early Industrial Revolution diminished only by the movement for and accomplishment of the Ten Hour Work Day followed by the same for the Eight Hour Work Day. The message from history is clear: the Six Hour Work Day will check and improve and make possible the benefits of the new Cybernetic Revolution - the marriage of computers and robots at the workplace.

Jobs. Analyze for yourself the lessons about them from the history of the transition from ancient chattel slavery, through the serf period, to wage-slavery today. The answer is again the same, Solidarity, worker with worker. It has brought us much to be proud of in terms of reigning in the horrors that are possible and it can bring more. The solutions to the current problems again will come only from worker to worker communion because it is still true that the boss is never your buddy.

Being a poet by nature, it is natural that one sees all things in a sense of cadence or as near perfect harmony as plausible and workably possible. There should be more producers not less and fewer parasites. To stress that point, I submit a bit of verse written by the author in 1957. It is copied, word for word, as it appeared then in The Searchlight. As a preface, let it be stated that, in order to present the message and not the man, I used pseudonyms in many forms: nom de plume, aliases, ghost writings for real persons, etc., any device that would promulgate the message:

OVERTIME AND UNDER PAY

Now don't be fools or bossmen's tools
And work your life away;
Defend what's right and change the rules
To lower hours and higher pay.

There's no excuse for the abuse
Of long and tiresome hours
That shorten life and further induce
The wealth of ruling powers.

For rich men's gold is bought and sold
In terms of human task
And they are brave, and they are

bold
When hours is all you ask.

Geneve Watkins

MAY DAY

by Floyd Hoke-Miller

Hooray, hooray, it's the first of May.
It's the workers' holiday.
Make it real, it's our heyday,
Demand shorter hours for labor!

It's May Day in the morning
It's when new life's a'borning.
It's when Nature is readorning
God's good Earth.

It's the harbinger of Spring,
It's the day for everything.
It's time to think, do and bring,
Honorable life to Labor!

This is the day that Providence
Has set aside sans pretense.
It rewardingly makes good sense
Stand up and use it!

Sample Batch File Using a Text File & Parameters

(Type in the following batch file, saving it as an ASCII file called EDIT.BAT. You also need EDLIN.COM.)

```
echo off
cls
if "%1" == "" goto ed
type ed.txt
pause
edlin %1
goto endbat
:ed
type ed.txt
echo WARNING: No filename was specified, exiting batch file.
:endbat
```

ASCII Text File

(Type in the following text file saving it as an ASCII file called ED.TXT in same directory.)

```
        THE EDLIN COMMANDS
COMMAND FUNCTION  DESCRIPTION
A       APPEND    ADD LINES TO
                  MEMORY
C       COPY      DUPLICATES LINES
D       DELETE    REMOVES LINES
#/#'s   EDIT      GIVE LINE TO EDIT
E       END       STOPS EDITING AND
                  SAVES FILE
I       INSERT    ADD LINE OF TEXT
L       LIST      DISPLAYS A RANGE
                  OF LINES
M       MOVE      MOVES A RANGE OF
                  TEXT TO GIVEN LINE
P       PAGE      BROWSE THROUGH A
                  FILE 23 LINES AT
                  A TIME
Q       QUIT      LEAVES EDLIN
                  WITHOUT SAVING
                  FILE
R       REPLACE   REPLACES TEXT
S       SEARCH    LOOKS FOR TEXT
T       TRANSFER  MOVES CONTENTS OF
                  ANOTHER FILE INTO
                  CURRENT FILE
W       WRITE     SAVES SPECIFIED
                  LINES TO DISK.
```

HISTORY OF COMPUTERS (Part II) The Computer As a Logic Machine

by Ronda Hauben

(Editor's Note: This is the 2nd part of a 4 part article that began in the last issue. The article grew out of a period of 2-1/2 years during which the writer taught computer programming at the UAW-Ford Employee Development Center at the Dearborn Engine Plant of the Ford Rouge Complex. Despite petitions and appeals to the UAW by Ford workers taking the classes, the classes were discontinued by Ford and the UAW. The underlying rationale for ending the classes seemed to be the notion on the part of Ford and UAW officials that workers do not need to learn to program. They only need to learn to run software. This view now seems to pervade computer education in Michigan. Programming classes have been cut out across the State and classes in using particular software programs have been substituted. The history of the computer, however, demonstrates how knowledge of computer programming is not only needed to successfully use computers at the current state of their technological development, but in addition, the broad dissemination of programming knowledge is needed to develop the uses of the computer. The *Amateur Computerist* newsletter grew out of the realization that we could not rely on companies or union-management sponsored training programs to support the dissemination of computer knowledge.

The article is accompanied by a BASIC (IBM version) program which is interspersed with the relevant sections of the article. Readers may want to type in the program and then run the indicated section as they read the article.)

```
600 PRINT "GEORGE BOOLE 1815"
610 PRINT "AN INVESTIGATION OF THE"
620 PRINT "LAWS OF THOUGHT 1854"
625 GOSUB 5000
```

George Boole was born in 1815. His parents could not afford to send him to the university. He taught himself Latin, Greek, and with his father's help mathematics. He went on to study the great masters in mathematics and put together an important theoretical treatise. In 1854 he published his work "*An Investigation of the Laws of Thought* on which are founded the Mathematical Theories of Logic and Probabilities."

```
630 PRINT "BOOLEAN LOGIC"
640 PRINT "1 IS EVERYTHING"
650 PRINT "0 IS NOTHING"
655 GOSUB 5000
```

Boole showed that the study of logic could be reduced to just two symbols "0" or "1". Everything is one. Nothing is zero. Boole's work in binary numbers and logic was to prove crucial in the building of computers a century later, by providing the binary basis of the modern computer.

```
660 PRINT "TRUTH TABLES"
665 PRINT
670 PRINT "AND TABLE"
680 PRINT "A","B","A AND B"
690 FOR A = 1 TO 0 STEP -1
700 FOR B = 1 TO 0 STEP -1
710 PRINT A,B,,A AND B
720 NEXT B
730 NEXT A
735 GOSUB 5000
```

Also, the ability of the computer to do And, Or, and Not logic operations is at the heart of how the modern computer functions.

```
740 PRINT
750 PRINT "OR TABLE"
760 PRINT "A", "B", "A OR B"
770 FOR A = 1 TO 0 STEP -1
780 FOR B = 1 TO 0 STEP -1
790 PRINT A, B, A OR B
800 NEXT B
810 NEXT A
820 PRINT
825 GOSUB 5000
```

In some computers, 1 stands for true and 0 for false. In Commodore and IBM, - 1 is true and 0 false. (In Apple, 1 will represent true, and 0 false.)

```
830 PRINT "NOT TABLE"
840 PRINT "A","NOT A"
850 FOR A = 0 TO 1 STEP 1
860 PRINT A, NOT A
870 NEXT A
875 GOSUB 5000
```

Over the course of the next hundred years there were numerous technological breakthroughs that helped pave the way for the development of a working computer.

```
900 PRINT "TECHNOLOGICAL BREAKTHROUGHS"
910 PRINT "THOMAS EDISON 1847"
920 PRINT "ELECTRIC LIGHT BULB 1879"
930 PRINT "HERMANN HOLLERITH'S"
940 PRINT "TABULATING MACHINE 1890"
950 PRINT "LEE DE FOREST"
960 PRINT "VACUUM TUBE 1912"
965 GOSUB 5000
```

These included the invention of the electric light bulb by Thomas Edison in 1879, the Tabulating Machine invented by Herman Hollerith to help with the census of 1890, and the invention of the vacuum tube by Lee de Forest in 1912. But it was WW II that provided a major impetus to produce a working computer.

THE WORKING COMPUTER

```
1000 PRINT "WW II"
1010 PRINT
1020 PRINT "COMPUTER NEEDED"
1030 PRINT "TO CALCULATE FIRING TABLES"
1035 GOSUB 5000
```

During WW II, the U.S. government found itself in desperate need of a computer. They needed to calculate the ballistic firing tables which gave gunners the necessary information to aim their weapons properly. Out of this need came the ENIAC, the modern parent of today's personal computers.

```
1040 PRINT "TEAM OF SCIENTISTS"
1045 PRINT
1050 PRINT "MAUNCHLY AND ECKERT"
1060 PRINT "BRAINERD AND GOLDSTINE"
```

```
1065 PRINT
1070 PRINT "ENIAC 1945"
1075 PRINT
1080 PRINT "KICKED OFF"
1090 PRINT "COMPUTER INDUSTRY"
1095 GOSUB 5000
```

To meet this wartime need, a team of scientists including Mauchly, Eckert, Brainerd and Goldstine received over $500,000 to carry on the task of building a working a computer. (*Bit by Bit*, p 124) It took a year to design the computer which they named the ENIAC and a year and a half to build it, and it wasn't finished until three months after the Japanese had surrendered. The project "kicked off the computer industry."

The ENIAC took up a whole room and it was made up of thousands of electronic parts.

How did you get it work?

"You did not sit down at a computer terminal and type in the instructions; instead, you set thousands of switches and plugged in hundreds of cables [like the cables on old telephone operator consoles] by hand, one at a time. In general, it took about two days to set up ENIAC to carry out a program."(*Bit by Bit,* Augarten, p128)

Augarten goes on to explain that an inadequacy of the ENIAC led to a particularly significant breakthrough: "The need to set ENIAC up physically was its most serious drawback, and the next step in the development of computers -- the one that led to the invention of the modern computer — was the constructing of a machine whose wiring didn't have to be touched and whose programs could be inserted and carried out automatically." (p 130)

```
1110 PRINT "EDVAC COMPUTER"
1120 PRINT "BEGUN 1944"
1125 PRINT
1130 PRINT "FIRST STORED PROGRAM"
1135 PRINT
1140 PRINT "JOHN VON NEUMANN "
1150 PRINT "LOGIC DESIGN CRUCIAL"
1160 PRINT "FIRST DRAFT....1945"
1165 GOSUB 5000
```

Firing tables hadn't required the ability to change the program, just the speed. Other uses for the computer did require changing the program, and thus rewiring the machine. So in October, 1944 the Ordnance Dept of the US Govt. issued a $105,600 contract for the EDVAC (Electronic Discrete Variable Computer). They were anxious to build a computer that would allow a stored program. Augarten explains the momentous significance of being able to store instructions that can be changed on the inside of a machine:

"Storing instructions in a computer... flies in the face of mechanical tradition. Machines have always been controlled from the outside; you turned some knobs and set some switches and the gadgets did your bidding. Even if the devices were controlled by punched tape or cards, like the Jacquard loom or the Analytical Engine, the best modus operandi seemed to be to stash the cards outside the machine and insert them when you needed them. Indeed, you couldn't very well hope to do more, since the practical shortcomings of gears and axles made it difficult if not impossible to build a large memory — although relays made the job much easier. As a result, no one, with the possible exception of Zuse [German inventor] conceived of the stored program before the invention of ENIAC." (p 133 - 134)

John von Neumann was born in 1903. He joined the ENIAC team, at Moore School in Philadelphia, in summer of 1944. Von Neumann understood that the crucial aspect of the computer was the logic of its design, not its electronics. In Spring, 1945, he wrote a 101 page summary which was the result of discussions among the team members about the logical basis of the computer called "First Draft of a Report on the EDVAC," explaining how the internal memory of the computer had to be stored separately from the input/output devices. His model was on similar conceptual lines as that worked out by Babbage.

```
1170 PRINT "PROGRAM IN MACHINE CODE"
1175 PRINT
1180 PRINT "1001001001100011"
1185 PRINT
```

```
1190 PRINT "SHORT CODE AND SPEED
CODING"
1120 PRINT "FORTRAN COMPILER 1957"
1125 GOSUB 5000
```

By the 1950's stored program computers replaced the ENIAC. The result was, instead of having to change a computer's wiring, you could feed instructions directly into a machine through punch card readers, magnetic tape decks, etc. Instead of physically having to rewire a machine, the operator could electronically change the pattern of binary switches. Thus a program would have had several lines of binary — like 10010000100 01001. A one meant that electricity flowed through the pathway, a zero that it didn't. By changing the program, one would change pathways of the electrical signals, and thus create a totally different outcome — much as rewiring a machine had achieved in earlier technology.

Typing in a lot of 0's or 1's satisfied the needs of the machine, but was very cumbersome for the operator. Programmers began working to try to streamline the task of writing programs. A team working on creating a language, put together a FORTRAN (Formula Translation) compiler by 1957. "Easy to learn and to use, it was an enormous advance over assembly languages like Short Code and Speedcoding." (*Bit by Bit,* Augarten, p 217)

Most of the computers of the 1950's and 1960's were built in the giant mold of ENIAC. "Only the largest institutions — universities, corporations, research institutes and government agencies — could afford them." As one writer says, "Big and costly, they were the very symbols of entrenched and centralized power — arrogant, haughty, impersonal, inefficient, and inaccessible." (*Bit by Bit*, p253)

Particularly, computers were inaccessible to the average person. John Kemeny, one of the inventors of the BASIC programming language describes the dilemma:

"For the first two decades of the existence of the high-speed computers machines were so scarce and so expensive that man approached the computer the way an ancient Greek approached an oracle.... A man submitted his request ... and then waited patiently until it was convenient for the machine to work out the problem. Only specially selected acolytes were allowed to have direct communications with the computer. In the original mode of using computers, known as batch processing, hundreds of computer requests were collected by the staff of a computation center and then fed to the machine in a batch."
(Quoted in *Bit by Bit*, by Augarten, p 253-254)

(To be continued...)

The *Amateur Computerist* invites contribution of articles, programs, graphics, money, etc. Send submissions to: R. Hauben, P.O. Box 4344, Dearborn, Mi. 48126. Articles can be submitted on paper or disk in ASCII format, (IBM or Commodore.) A one year subscription (4 issues) costs $5.00(US). Add $2.50 for foreign postage. Permission is granted to reprint any article herein, provided full credit, including subscription information is given and the reprinting is not for profit.