THE WHY AND HOW OF COMPUTING IN THE SECONDARY SCHOOLS

F. J. Gruenberger

September 1963

P-2768

THE WHY AND HOW OF COMPUTING IN THE SECONDARY SCHOOLS

F.J. Gruenberger*
The RAND Corporation, Santa Monica, California

For a comprehensive discussion of the role of the computer in the high school curriculum, we should, of course, consider more than Why and How. We might go into the What: What should we try to teach about computers and with computers? And we might also toy with the Who: Who should do the teaching and to what groups of students?

But let's take these things in order; the first one, logically, is Why.

The question is not why we should teach computing. We must assume that knowledge of computing and computers is good and that large numbers of people must be trained and educated each year from now on. The question is, do we want this training and education to begin prior to college? It is obvious that many of us believe that computing belongs in the high school, but we should examine the reasons. Many subjects are competing for time, space, and money in our school systems. It can be argued that if computers come in, something else might have to leave. If this is so, there had better be powerful reasons for what we are proposing.

The first argument is common to many subjects. It goes like this: The pace of technology is speeding up,

---

*Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of the RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced as a courtesy to members of the staff.

This paper was prepared for presentation before the National Council of Teachers of Mathematics, University of Oregon, August 23, 1963.

and young people will emerge into a complex society, and we should accelerate their training in the lower grades. Essentially, this says "We're in a hurry, so let's cram faster." Thus we see calculus moving from the 14th grade toward the 12th (though admittedly only for the brightest students). We also see the same argument applied to driver training courses. It can be argued that driver training is literally training for survival, and perhaps my survival, which touches me. The same argument, if it is valid, applies to computing, and perhaps we could make a case for computing being a more basic tool (and of wider application) than the calculus. I would prefer to restrict the discussion to "Where should computing be taught." I am trying to point out by analogy that we should always question new demands on the time or money of either the school or the pupil. I believe that computing as an academic subject can stand on its own. It is not a fad and it is becoming a vital part of almost everyone's education.

For any subject, there is a threshold point, it seems to me, below which you cannot go. Thus, you can't teach algebra to five-year-olds, no matter how bright; their minds just are not mature enough to absorb the symbolism and abstraction. I would seriously question the efforts to introduce the calculus to the 17-year-olds. It is easy to demonstrate the learning of the formulas and mechanics; I am questioning the learning of the content and its meaning. We must be able to demonstrate that more understanding has taken place than could be exhibited by, say, a tape recorder, or a parrot. So in advocating computing as a subject for the 12th grade, I must determine where the threshold is.

Admittedly, for any given subject, the threshold is different for different people. The subject we are

considering, though, is proposed for mass education; we must worry about the threshold point for all the students who might arrive at the fourth year of high school mathematics. Many experiments have established that fair understanding of computing can be realized with gifted 12-year-olds (i.e., 7th or 8th graders). It seems reasonable to assume that we would be well over the threshold at the 12th grade. With computing, we can prove that understanding has developed in the child (and proof is difficult with every other subject). When a student can take a real computer problem and carry it through all its stages independently (namely, analysis, flow-charting, coding, debugging, testing, and production), then there is little question but that he has mastered much of the subject. We can insure that the problem he works on is, indeed, a good computer problem by having him select from a limited--but very large--set. If so, we have at least established feasibility.

The threshold concept is not new. Consider the problem that a mother cat faces in giving her kittens the training they need for survival. The lessons she gives, in stalking and pouncing, and in how to go through a doorway, are basic and vital; the kittens who fail to learn stand a good chance of not becoming grown-up cats. The first thing she does (by instinct, to be sure) is wait for them to reach the minimum age--the threshold for the course she is about to give. It comes at about four weeks of age, and it would be futile to teach it earlier. She then teaches by example; for instance, by going through a doorway in the time-tested manner. They then learn by doing. Usually they do it all wrong--once. They are promptly punished, by being knocked five feet or so by mommy's big paw. They get one more chance, and on the second try they get it

right, and are promptly rewarded with lunch. This is known as the Felix Domesticus school of teaching. Cats don't seem to need any more modern methods, since this one works splendidly. I have digressed; the important point is that threshold effect.

But now I argue that the learning of computing is analogous to the learning of a foreign language. In fact, the first third of a course in computing is the learning of a foreign language, and moreover is conducted in that language. It is well established that foreign languages are best learned young (in the case of languages, the younger the better. In some parts of the world--China for example--small infants learn idiomatic Chinese readily). For aged people, no amount of time and effort seems to be enough to learn a foreign tongue without an accent and with full command of the word order in a sentence.

This, to me, is the strongest argument; namely, that it's simply easier and more fun to learn this subject in high school than in college. Now we can pile on all the other arguments:

a) The high school student has less preconceptions and misinformation that will only have to be erased. He tends to bring a blanker slate to write on, in other words.

b) The college-bound student who has had some computer training in high school has a distinct leg up on his arrival on campus. It is becoming very difficult these days to find a college that is computer-free.

The day is not far off when the roles of the entering freshman and the college admittance officer will be somewhat reversed. The student will be asking, with justification, whether he should put this college on his list of acceptable institutions. He will ask how much computing capacity the college has, and how much

of it, per week, he can get.

c) The computer can be used as a teaching device to make the learning of other things (e.g., mathematics) easier, or more attractive. There is also a prestige factor and a reward factor to consider in having a computer course available in the senior year.

Those of us in computing claim--and believe--that the learning of computing has a beneficial effect in regard to clear, logical organization of one's thinking on new problems. We are claiming the transfer effect, and we have, to date, little or no evidence to support it. I recall (in slightly different context) the same claim made some years back for the learning of Latin; namely, that besides all its other advantages, it was good for the learning of English. I think the transfer argument is as weak for computing as it was for Latin; at least, we should demand proof that the transfer actually takes place. The best I can say right now is that I think it has made my thinking more orderly--but you can see what an illogical argument that becomes.

So much for Why. I have already tacitly implied the Who: The computer course should be embedded in the mathematics department. Specifically, if it is to be in the regular curriculum, I think it should be in the trigonometry course. A significant portion of a traditional trig course deals with the mechanics and dog work of computation, and the computer can pay back most of the time it takes to teach its fundamentals. We can hope, of course, that the computer will not be labelled "exclusive property of the mathematics department," but that its use be extended to other departments. As an example of what I mean here, all the teachers have common chores of bookkeeping in connection with grades. The computer might offer some relief from such chores

as a starter; eventually its power can be extended to more sophisticated uses. The foreign language teacher, for instance, might easily be lured into trying a machine approach to word-for-word table look-up translations of simple texts. The teacher of business practice should be attracted to some of the simpler business applications of the computer, and so on. In addressing the NCTM on the subject of What, I discussed what I felt were three important areas:

1) Traditional problems. We can use the computer to solve many problems that have always been included in secondary math classes. There was mentioned the solution of triangles (i.e., all the stock trigonometry problems); there is also the solution of systems of equations, and so on. Of great importance, along these lines, is the topic "What should we not compute?"--in other words, what problems are not good computer problems? Let me cite two examples of what I regard as not-good computer problems:

   a) The calculation of 17^5, which is not only trivial, but a one-shot problem. A good computer problem demands repetition.

   b) The playing of tick-tack-toe. This can be great fun, and the programming for it can be challenging, but it's the wrong thing to compute. Every possible board situation in the game can be stored in any computer, and the correct move can simply be looked up. The world is jammed with good and real computer problems; why allow students to waste their time and the machine's time?

2) Advanced problems. There are many problem areas in advanced mathematics--calculus, number theory, college algebra, analytic geometry--which can be successfully tackled with a computer, if only by cut-and-try methods.

For many problems, it is possible to show that the problem is not yet solved--and the student may be the one to show us--or that the known solutions are limited and he can move the boundaries of knowledge back a bit. This is the most challenging motivation I know for the good student. By making a solution feasible, the student is led to inquire into the analytic methods. Inquiring students are always good, sometimes almost as good as inquiring teachers. In my experience, the computer acts as a catalyst to produce both.

3) Entertaining problems. The whole field of computer work is somewhat akin to a large jigsaw puzzle and it seems to attract the same kind of addicts. The world is full of puzzles that are ideal for computer attack. Everyone's experience along these lines has been the same; namely, that if you furnish a computer to bright kids, you need only stand back and let them go. High school students will not tolerate any waste of the machine's time, nor any work on trivial problems. They will demand answers to their questions (does the idea of students demanding to learn more intrigue you?). The high school teacher who finds all of this slightly frightening need not worry. He has several things in his favor:

a) He has several years to learn the subject, and has always the assurance--as with any subject--that lots of stupider people have done it before him.

b) His students will learn the subject at high speed, with or without him, and will be delighted to show him the fine points.

c) As with no other subject, he can respond to most questions with "Why don't you try it and see?" You can't hurt a computer through its console buttons, and there are no mysterious hidden elements. Both the teacher and the student can personally check out every topic.

With any sort of luck, they can find a better way to do many things. And that brings me to the question: How. How do we couple our students to these wondrous machines? It is well known that the machines are frightfully expensive.* I would suggest that many schools are unaware that they've been moving in the wrong direction. It isn't really necessary (however desirable it would be) to move a machine to the students; why not move the students to the machine? It's a rare school these days that isn't within 10 miles of a computer, and I have yet to hear of a computer user who wouldn't encourage an earnest group to visit his installation and use some machine time. In most such cases, the school will acquire a free teacher for lagniappe, and one who works long hours with great enthusiasm. It's not the ideal solution, but it's a fine way to operate today. Maybe tomorrow we can get the manufacturers to make an inexpensive machine for the schools. The manufacturers are understandably shy about an undemonstrated market, but, on the other hand, when the market develops (there are 11,000 high schools in the country of 1000 or more students), they want to be there with the goods. Meanwhile, I should sound a warning against inexpensive non-computers, of which there are already too many on the market. Don't waste your precious dollars on paper clip gadgets that purport to be "just as good" as a computer. Everyone is well aware that adding the word "programmed" to the cover of a book doesn't automatically make it into a programmed text. Similarly, no matter how big the word "computer"

*There is such a thing as a "free" machine, occasionally, but beware of it, since it may be a white elephant. The cost of maintenance may be prohibitive, and the "free" machine is usually old and unreliable. No one sets out to teach the use of an inoperative machine.

appears on the nameplate or the brochures of a cheap gadget, that doesn't make it a computer, or a computer simulator, nor a device to "do the same things as the giant electronic brains" (sic). A computer is an internally programmed calculator; so far, all of them are electronic. The secondary schools shouldn't settle for less, particularly when there is probably access to one nearby for nothing.

The real computers <u>are</u> getting cheaper, and quite rapidly. Coincidentally, textbooks are appearing that are designed for high school use, with every promise of many more to come. The only real shortage right now seems to be trained teachers. That, of course, is the When as it relates to you. If you have not already begun to learn computing, what are you going to be doing next summer?