The New York Times Mag

WHEN THE COMPU

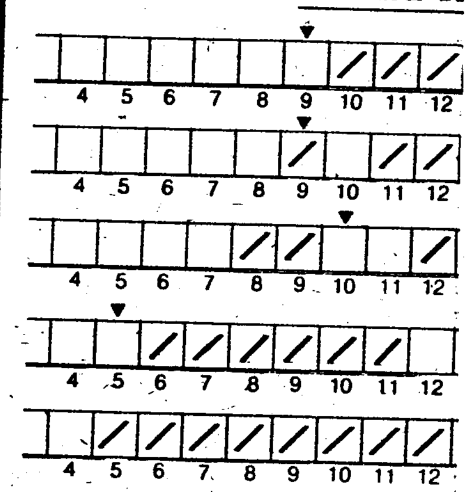
Its most profound im
its usefulness as calculating ma
self-replication –

By Jeremy Bernstein

In the spring of 1955, I completed my Ph.D. thesis for the Harvard physics department. I had done a theoretical problem, my principal memory of which is that it required the computation of 75 numerical integrals. An integral, for the nonspecialist, is simply the area under a given curve. Each integral took me something like two days to compute, and the 75, nearly six months. Some elementary integrals could be looked up in books,* but mine could not. I had to calculate them myself. The mind boggles now at how I once spent six months of my life—and not unhappily: Sisyphus, as Camus pointed out, was basically a happy man.

Just as I was finishing this task, there appeared on the scene a whiz kid from Cal Tech. He was about to begin a post-doctoral appointment at M.I.T., and it turned out, much to my amazement, that he had done essentially the same problem with two notable differences: (1) His basic theory was better, and (2) he had cultivated a group at M.I.T. that had built one of the first electronic computers—a vacuum-tube affair—and, though a dinosaur by present standards, it enabled him to calculate each of his integrals in about two minutes. I decided that in the future I had two choices: I had either to learn to deal with electronic computers, or to avoid numerical integrals. In the past 20 years, I have, by and large, chosen the latter alternative.

TURING M



Stages of a program (a sequence of one square right or left and to p multiplying 4 times 2 on a theoretic by Alan Turing (below). Thus began tronic self-programming computer p believe it may be possible to add a bi

I bring all of this up because a number of morals can be drawn about man and the machines he has created. In science, theory and experiment interrelate but, generally speaking, an experimenter begins from some sort of theoretical premise. For example, in high-energy physics these premises can take such a simple qualitative form as: If the laws of nature were symmetric between particle and antiparticle, then the neutral-pi meson could decay only into an even number of light quanta. Don't panic, if that terminology is unfamiliar. The point is: This is a statement that could be—indeed *was*—made without aid of any computer. Even in the case of theoretical statements involving numerical work done by computer, the theoretical structure is always developed by a physicist and not the computer. Computers do not create theories of physics. And no paper in theoretical physics, at least until now, has required the use of a computer to be understood.

In a typical modern high-energy experiment, millions of events occur—photographed tracks in a bubble chamber, or photographed spark discharges, for example. Experimenters make extensive use of computers in recording and analyzing these events. It is not uncommon for an experimenter to prepare a plot on which each point itself represents a million events. Obviously, this kind of data processing must be done by computer. But for the results to be comprehensible, they must be fitted by some sort of curve that arises from a humanly created theory. The chain is human to human with the machine somewhere in between. It is sometimes said that the computer is to this process as the microscope is to vision. I think this analogy is flawed. Forty-eight people staring at a glass of water with the unaided eye will not see the microbes swimming around in it; while one person with a microscope will see them. The microscope has revealed somthing that was,

*The numerical integrals published then dated back to the Depression, when otherwise unemployed mathematicians were engaged by the W.P.A. in calculating them by hand — that is, with the aid of mechanical calculators. The better old desk calculators could add, subtract, multiply and divide. Their gears were run by electric motors; each step had to be performed in order, and all intermediate results had to be tabulated. These instruments seem to have taken their place in nostalgia along with the Packard.

'HE COMPUTER PROCREATES

Its most profound impact arises not from
ɘfulness as calculating machine, but from its potential for
self-replication—even mutation.

mpleted my Ph.D. thesis for the Harvard
done a theoretical problem, my principal
required the computation of 75 numerical
he nonspecialist, is simply the area under
ɔk me something like two days to com-
onths. Some elementary integrals could
could not. I had to calculate them myself.
nce spent six months of my life—and not
ted

isk,
hiz
to
at
my
en-
two
asic
had
had
om-
nd,
nd-
ach
tes.
wo
eal
oid
20
the

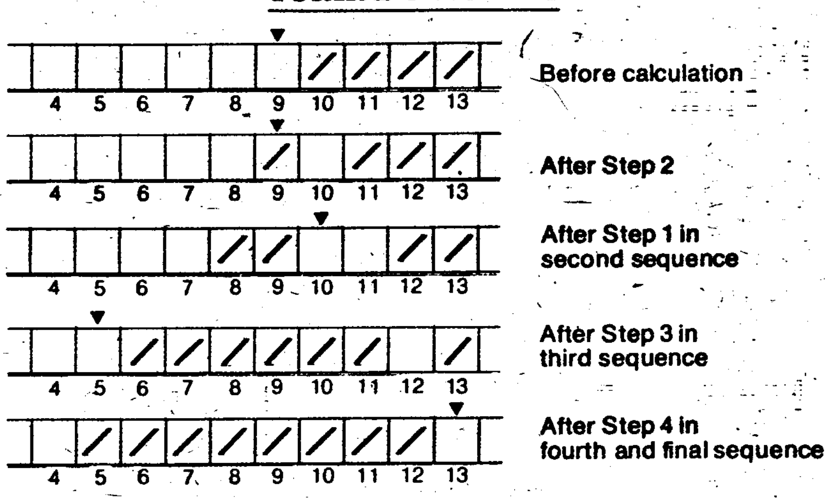
im-
an
In
er-
an
rt of theoretical premise. For example, in
can take such a simple qualitative form as:
tric between particle and antiparticle, then
only into an even number of light quanta.
unfamiliar. The point is: This is a statement
le without aid of any computer. Even
nents involving numerical work done by
e is always developed by a physicist and
not create theories of physics. And no
least until now, has required the use

rgy experiment, millions of events occur
le chamber, or photographed spark dis-
ers make extensive use of computers in
ts. It is not uncommon for an experimenter
point itself represents a million events.
ssing must be done by computer. But for
they must be fitted by some sort of curve
ed theory. The chain is human to human
tween. It is sometimes said that the com-
croscope is to vision. I think this analogy
g at a glass of water with the unaided eye
ng around in it; while one person with a
croscope has revealed somthing that was,

in principle, invisible to the human eye. On the other hand, there is nothing the computer does in this chain that could not be done by enough human brain power. Although it took me six months, I did finally do the numerical integrals.

Recently I had the opportunity of discussing this matter with Arthur C. Clarke, whose fictional computer HAL—invented in collaboration with Stanley Kubrick for the film "2001"—could do about *anything*. Indeed, HAL could do too *much* of anything. Clarke made the point that electronic computers can now do computation that would require the whole human race working together to accomplish without them. (Perhaps a more estimable activity than many others the race often engages in.) Furthermore, the speed of the individual operations on the computer is something totally beyond human capability; its basic calculations can be completed in the millionths of a second or less. Still, in the man-machine-man process, the machine in the middle can, at least in principle—and, at least, at present—be replaced by humans. Clarke once wrote a story entitled "Into the Comet," in which a spaceship's computer fails and, lacking the proper orbit, the ship heads for disaster. A crewman, of Japanese origin, teaches the entire crew to make abacuses with wires and beads and the ship is saved.

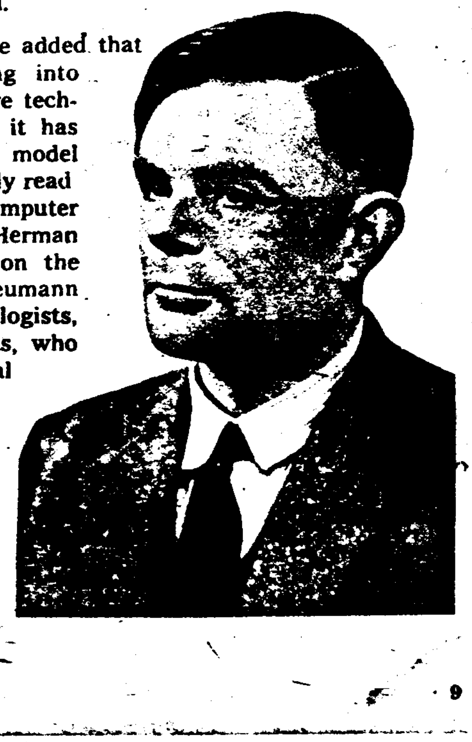
TURING MACHINE



Stages of a program (a sequence of directions to the arrow to move one square right or left and to print a slash or erase one) for multiplying 4 times 2 on a theoretical computer invented in the 30's by Alan Turing (below). Thus began development of the modern electronic self-programming computer process, to which scientists now believe it may be possible to add a biological component.

The computer has quantitatively enlarged the sort of calculations and experiments an individual scientist can take on in his lifetime but, as far as I can tell, it has, on its own, created nothing. In this respect, I am reminded of my one and only encounter with the great Hungarian-born American mathematician John von Neumann. It was von Neumann who developed the theory of stored programming—that is, the capacity of a computer to modify its own instructions as a computation unfolds. He delivered a series of lectures at Harvard while I was an undergraduate, and I was enormously impressed. After one lecture, I found myself in Harvard Square alongside the great man himself as he hurried by to find the subway. I thought, correctly as it turned out, that this would be the only chance I would ever have to ask him a question. I seized the occasion. "Professor von Neumann," I asked, "will the computer ever replace the human mathematician?" "Sonny, don't worry about it." Literally, that is what he answered.

All this having been said, it must be added that the computer has injected something into modern scientific thinking beyond mere technology. For the first time, I believe, it has presented us with a machine-tooled model—still primitive—of ourselves. I recently read in a history of computers ("The Computer from Pascal to von Neumann," by Herman Goldstine) that in his early papers on the logical design of computers, von Neumann took his notation from two physiologists, Warren S. McCulloch and Walter Pitts, who were trying to make a mathematical model of the human nervous system. Von Neumann was trying to create what might be *(Continued on Page 34)*



ed then dated back to the Depression, when were engaged by the W.P.A. in calculating of mechanical calculators. The better old desk y and divide. Their gears were run by electric in order, and all intermediate results had to be have taken their place in nostalgia along with

Jeremy Bernstein is a professor of physics at Stevens Institute of Technology and a staff writer for The New Yorker. His latest book is "Einstein."

Computer

Continued from Page 9

described as an electronic nervous system. Now, clearly it would be a fatal mistake to try to construct a model of the nervous system by working from the outside in. That is to say, if you take as your primary data products of the nervous system such as Einstein's theory of relativity, Beethoven's Ninth Symphony and Van Gogh's "Starry Night" and, from these, attempt to deduce the construction of the apparatus that produced them, you are not likely to get far. It would be like trying to deduce the structure of the elementary particles of subnuclear physics by contemplating Mount Everest. The idea, rather, is to put together a vast array of very primitive objects and to see what such an array working in concert can produce. The fundamental components of the McCulloch-Pitts model, described in a celebrated paper entitled, "A Logical Calculus of the Ideas Imminent in Nervous Activity," were "neurons" connected by wires that could transmit electrical pulses. (In the human brain there are about ten billion neurons—organic molecules about a hundred-thousandth of a centimeter in diameter—wired together by axons, or fibers, that can be several feet long.) For purpose of the analysis a neuron acts as a relay station for electrical pulses. If such a station receives a sufficiently strong impulse, it will "fire," or emit a pulse. If these neurons are wired together, in units, things can be arranged so that it takes the activation of, for example, a pair of neurons to fire a third one and so on. That event in this so-called "logic circuit" might be described, in the language of formal logic, as *A plus B implies C*. Now, it has been known since the pioneering work of Bertrand Russell and Alfred North Whitehead that even the most complicated mathematical statements can be broken down to a collection of such primitive logical propositions. Hence, McCulloch and Pitts were emboldened to conclude: "Anything that can be exhaustively and unambiguously described, anything that can be completely and unambiguously put into words is, *ipso facto*, realizable by a suitable finite neural network." In other words, neural networks can carry out the processes of mathematical logic.

Von Neumann was deeply impressed by this analysis, since the computing machines he was designing were essentially neural networks with electronic devices — vacuum tubes in the original, and now archaic, versions of the machines—playing the role of the organic neurons. These machines, therefore, could in principle do anything that a McCulloch-Pitts model could do. Being the kind of genius that he was, von Neumann did not leave the analysis there. He built on the work of a remarkable young British mathematician, Alan Mathison Turing, whose work remains largely unknown, except to specialists. Yet it may turn out that, when a future historian of automation looks back at the really revolutionary implications of the so-called computer revolution, these will have much more to do with the as yet unrealized abstract ideas of Alan Turing, as generalized by von Neumann, than all of the new airline reservation systems put together.

Alan Turing, whose life is described in a moving book written by his mother in 1959, died five years earlier at the age of 42, perhaps by suicide. From 1936 through 1938, he studied at Princeton, where his work came to the attention of von Neumann. (Some of the work was done independently by the American logician Emil L. Post, but apparently von Neumann was not aware of it.) Von Neumann offered him a position as his assistant at the Institute for Advanced Study. Turing declined it, preferring to return to Kings College in Cambridge, England, where he was a fellow. He later worked on the construction of the first British computers.

Turing invented what is now known as the Turing machine, which is actually not a real machine at all but, rather, an abstract construct —an idea—for an apparatus that could be instructed to make mathematical calculations. There are various ways of describing the basic idea. But as Mark Kac, a professor of mathematics at Rockefeller University, has put it, the Turing machine has an infinitely long tape divided into identical-sized squares, each one of which either is blank or contains a slash. Over the tape, there is a movable arrow. The machine can be given four basic directions, denoted: L, R, ° and /. The L means move the arrow one step to the left; R, one step right; ° means erase the slash; / means print slash.

The machine may be programmed to carry out a sequence of operations. Assume for the sake of the discussion that your Turing machine had four slashes on your infinitely long tape, one each in Squares 10, 11, 12 and 13. To get it to double those four, in other words to get it to multiply 4x2, you first move the pointer to Square 9, and then issue a sequence of instructions, bearing in mind that at each step the machine has to be given two alternative courses of action: (1) R (that is, move the pointer one square to the right): if blank repeat Step 1 (that is, move another square to the right), or if not blank erase / and go to Step 2; (2) L: if blank print / and go to Step 3, or if not blank reprint / and repeat Step 2; (3) L: if blank print / and go to Step 4, or if not blank reprint / and repeat Step 3; (4) R: if blank leave blank and go to Step 1, or if not blank reprint / and repeat Step 4. Now this entire sequence is repeated four times, after which the answer appears as a series of eight slashes, in Squares 5 through 12. (Unfortunately, for the pointer and the machine's imaginary fuel supply, there is no way in this early Turing program to stop the pointer; after completion of the fourth series, it keeps encountering blank squares, leaving them blank and moving one space to the right—on to eternity!)

All this may seem a bit primitive but Turing went on to prove a most remarkable theorem: that you could also construct a general-purpose machine—he called it a universal machine—on whose tape you could write any number of programs, or codes made of slashes and blanks, and that the universal machine would read these instructions and carry them out. (In Turing's time, it was believed that a universal machine would have to be enormous and have to be given millions of instructions, but now the theory has been greatly simplified. Prof. Marvin Minsky of M. I. T. holds the record for building the smallest universal machine: It has 28 instructions.) In Professor Kac's words: "The Turing machine owes its fundamental importance to the remarkable theorem that all *concrete* mathematical calculations can be programmed on it . . . In other words, every concretely stated computational task will be performed by the machine when it is provided with an appropriate, finite set of instructions." (Turing also showed that problems exist for which no program can be devised in principle. These are the computing machine analogues of the celebrated undecidable propositions of mathematics first studied by the logician Kurt Gödel. The machine is, in this respect, no better or worse off than the human mathematician.)

Now for the great leap forward. Von Neumann asked himself whether programs could be devised that would instruct the Turing machine to reproduce itself. It had always been supposed that machines were used to produce objects that are less complicated than the machines themselves, that only biological reproduction transmitted total complexity or, indeed, through mutation, increased the complexity. A machine tool, for example, by itself cannot make a machine tool. One must adjoin to it a set of instructions and these usually take the form of a human operator. Hence, the complete system is the machine tool plus the human operator. Clearly, this system will, under normal circumstances, produce a machine tool minus the human operator—hence vastly less complex.

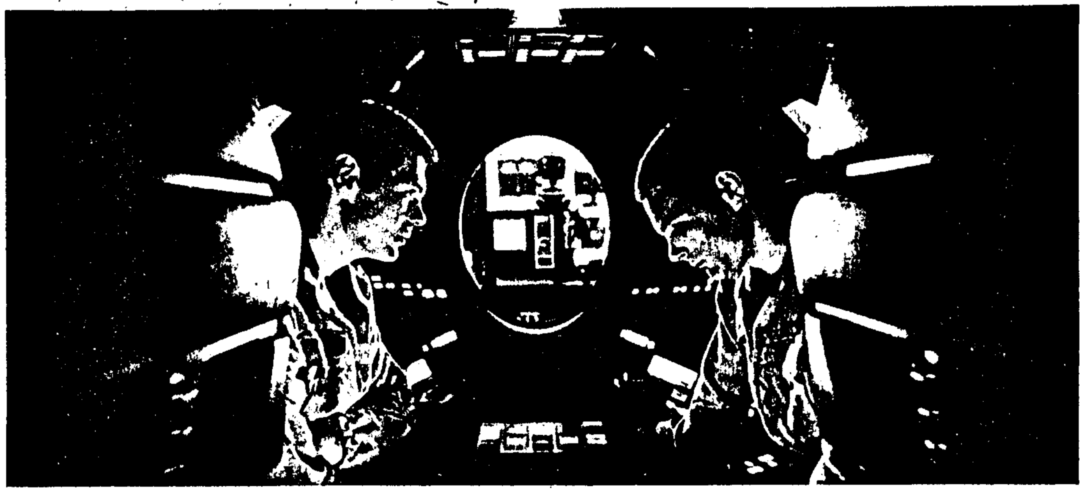
Von Neumann discussed these matters in the Vanuxem Lectures at Princeton in 1953, and since then these talks have acquired an almost legendary character. They were never fully recorded, however, though fragments appeared in a book entitled "The Computer and the Brain," and they were discussed in 1955 in an article in Scientific American by John G. Kemeny, now president of Dartmouth.

When we discuss self-replicating machines we must be clear about the ground rules. In Kemeny's words: "What do we mean by reproduction? If we mean the creation of an object like the original out of nothing, then no machine can reproduce, but neither can a human being. . . . The characteristic feature of the reproduction of life is that the living organism can create a new organism like itself out of inert matter surrounding it.

"If we agree that machines are not alive, and if we insist impossible. We must omit the word 'living.' We shall ask that the machine create a new organism like itself out of simple parts contained in the environment."

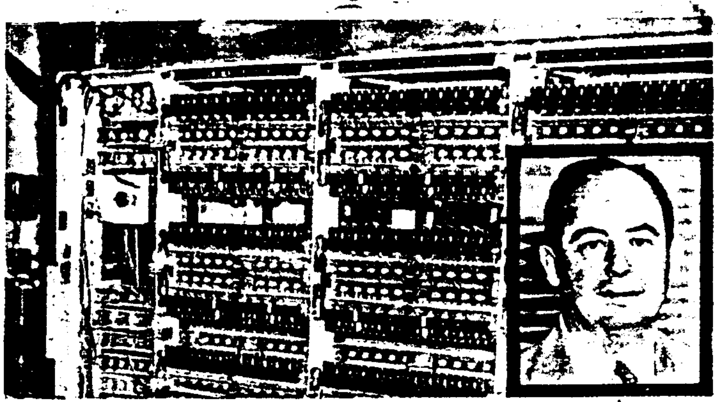
Von Neumann showed as early as 1948 that any self-replicating apparatus must necessarily contain the following elements. There must be the raw materials. In his abstract example, these are just squares of paper—"cells"—waiting around to be organized. Then we need the program that supplies instructions. There must be a "factory"—an automaton that follows the instructions and takes the waiting cells and puts them together according to a program. Since we want to end up with a machine that, like the original, contains a blueprint of itself, we must have a duplicator, a sort of Xerox machine that takes any instruction and makes a copy. Finally, we must have a supervisor. Each time the supervisor receives an instruction it has it copied and then gives it to the factory to be acted on. Hence, once the thing gets going it will duplicate itself, and, indeed, von Neumann produced an abstract model containing some 200,000 cells, which theoretically did just this.

Those who have had some education in modern genetic theory may have heard bells going off, or had the sense of *déjà vu,* upon reading the above description. It could apply as well, in abstract outline, to biological reproduction. We are, by now, so used to the idea of computer analogies to biological systems that they may appear obvious. One must keep in mind that they were not obvious, at all; indeed, they are only a few decades old. Von Neumann's analysis was five years ahead of the discovery of the double-helix structure of DNA, and preceded by several more years the full unfolding of what is called the "central dogma" of genetic replication. In a Vanuxem Lecture in 1970, Freeman Dyson of the Institute for Advanced Study made a sort of glossary translation from von Neumann's machine to its biological counterpart. The "factory" is the ribosomes; the copying machine is the enzymes RNA and DNA polymerase; the supervisor is the repressor and depressor control molecules, and the plan itself is the RNA and DNA. Von Neumann was there first.



Computer amok: In the film "2001," the computer HAL (one of its components visible through the porthole) spies on the astronauts (Gary Lockwood and Keir Dullea) it seeks to destroy.

His early training in Budapest was as a chemical engineer, and he never lost his feeling for engineering practicalities. He was not content to think purely in the abstract. Hence, he raised the following question: Real automata, including biological ones, are subject to error. There is a risk of failure in each of the basic operations—a wire can come loose. How can one design a system that will be reliable even if the basic operations are not completely reliable? The secret was *redundancy.* Suppose, to take an example from Goldstine, one has three identical machines, each of which makes a long calculation in which each machine makes, on the average, 100 errors. The way to improve reliability is to connect the machines, and require them to agree on one step before they go on to the next. If the system were set up so that, once two machines agreed, they could set the third at the agreed value and then proceed, then it turns out that the chance of error would be reduced from 1 in 100 to 1 in 33 million! Von Neumann concluded that the central nervous system must be organized redundantly to make it function at a suitable error level. This conclusion also appears to be correct. Von Neumann realized too, that if the universal Turing machine could be made to reproduce itself, it could evolve. If the program was changed, say, by "mutation," and this change was such that the machine could still reproduce,



Early computer (circa 1945): Developed by Princeton's John von Neumann (inset), it could modify its own instructions. He then posed the question: Can it self-replicate?

it would produce an altered offspring.

In Freeman Dyson's words, "Von Neumann believed that the possibility of a universal automaton was ultimately responsible for the possibility of indefinitely continued biological evolution. In evolving from simpler to more complex organisms you do not have to redesign the basic biochemical machinery as you go along. You have only to modify and extend the genetic instructions. . . . Everything we have learned about evolution since 1948 tends to confirm that von Neumann was right."

Where does all of this leave us? The point I have been trying to make is that, as far as I can see, the most profound impact of the computer on society may not be as much in what it can do in practice, impressive though this is, as in what the machine is in theory, and less to do with its capacity as calculator than with its capacity for self-replication.

Recently, I discussed these matters with Professor Minsky, who is one of the foremost authorities on machine intelligence. He told me that von Neumann's complex arguments have now been greatly simplified by his successors. Abstract models of self-reproducing machines have been devised that are extremely simple. Moreover, real computing machines are now almost self-replicating. One uses a computer to program the design of a computer and this design is given to a computer that supervises the actual physical construction of the new computer. One must supply from the outside the actual silicone chips on which the circuitry

now seems conceivable—in principle, at least—that perhaps with the addition of some primitive biological components (who knows what!), the process can be further developed to the stage where self-reproducing automatons can be made that are compact and, acting in concert, can do just about anything. In his Vanuxem Lecture, Dyson gives several examples of what colonies of these machines might accomplish, for good or evil, if let loose on earth or in outer space — such as bringing vegetation, light and heat to Mars. With a little thought the reader can supply his own examples. For some reason, as admiring as I am of the logic of automatons, I find the prospect chilling.

I suspect—and this is also emphasized in Dyson's lecture —that for self-reproducing machines to do anything interesting they must have a high level of interorganization. As Dyson put it: "The fully developed colony must be as well-coordinated as the cells of a bird. There must be automata with specialized functions corresponding to muscle, liver and nerve cell. There must be high-quality sense organs and a central battery of computers performing the functions of a brain," which may mutate and proliferate. In time, we may no longer recognize them. In this respect, Sara Turing quotes a letter in her book about her son that she received from the wife of one of Turing's closest colleagues, M.H.A. Newman. Mrs. Newman wrote: "I remember sitting in our garden at Bowdon about 1949 while Alan and my husband discussed the machine and its future activities. I couldn't take part in the