ham, a young astronomer who later became professor at Berkeley; and the author. Its mission was computing machines in general, but its emphasis was on ENIAC. It was 1945; the war was still on, though obviously drawing to a close; the Proving Ground was as busy as ever developing and testing weapons and ammunition; but ENIAC was still a jumble of components, in no position to help in the daily business of computing; and the Computations Committee was reduced to working on a few isolated problems, some real, some only for testing. It was 1946 before even test problems were actually put on the machine.

Two years later I found, in a file cabinet at the Ballistic Research Laboratories at Aberdeen, a copy of a letter describing one of the first problems ever run to completion on an electronic computer. The letter, addressed to the commanding officer, was from D.H. Lehmer, who had programmed the problem and run it on ENIAC, with J. Mauchly serving as "computer operator," during the three-day weekend of July 4, 1946. The running time of the problem occupied almost the entire weekend, around the clock, without a single interruption or malfunction. It was the most stringent performance test applied up to that time, and would be an impressive one even today. The problem was only a "test problem" from the point of view of the Army, but it provided an intrinsically important result in the theory of numbers. The cost was negligible since, as mentioned, the machine would have been left turned on in any case.

One of the peculiarities that distinguished ENIAC from all later computers was the way in which instructions were set up on the machine. It was similar to the plugboards of small punched-card machines, but here we had about 40 plugboards, each several feet in size. A number of wires had to be plugged for each single instruction of a problem, thousands of them each time a problem was to begin a run; and this took several days to do and many more days to check out. When that was finally accomplished, we would run the problem as long as possible, i.e. as long as we had input data, before changing over to another problem. Typically, changeovers occurred only once every few weeks. In between we had to cope with malfunctions of the machine, usually due to dead or submarginal tubes. A faulty tube could be replaced in minutes, but it might have taken days to locate it.

It was S.O.P. to run every problem at least twice, for checking, and to run test cases at frequent intervals. Between runs there were waiting periods for inspecting results, punching cards, and miscellaneous headscratching, since it was impossible to switch problems on short notice. In all, there were periods when not more than 5 percent of all available machine time resulted in useful work.

By 1947 a solution to these troubles appeared on the horizon. In retrospect, it seems to have been a forerunner of what we now call higher-order programming languages. The idea was to encode the instructions of a problem on the "function tables," three panels of the machine, each of which bore 1,200 ten-way switches. They had been intended as a computer-accessible table lookup, e.g. for empirical functions, but it was now proposed to use them to set up the succession of instructions, each represented by a two-digit number. The wiring of the plug boards would be set up permanently on the machine in a way that would cause the machine to read a number from the table, carry out the instruction encoded by it, go on to reading the next number, etc. Thus, the background wiring played the role of a present day "compiler"—more specifically, of an interpretive routine, since the source code had to be read and interpreted anew for each run, and no permanent object code was set up. This mode of operation would slow down the machine, of course; it was estimated that its speed would decrease at least by a factor of 5, a small price to pay for eliminating the long set-up time. We were, in effect, using ENIAC to "simulate" the future stored-program computers, which were then still on the drawing board.

Several versions of background wiring and their corresponding source languages were under discussion, each having a vocabulary between 50 and 100 instruction types. Their implementation and testing began in 1948. They were still only on paper at the end of 1947, when the Association for Computing Machinery was founded and held its first national meeting at Aberdeen Proving Ground. The attendance at that meeting was 300; the program consisted of about a dozen papers. We had succeeded in obtaining John von Neumann as keynote speaker. He discussed the need for, and likely impact of, electronic computing. He mentioned the "new programming method" for ENIAC and explained that its seemingly small vocabulary was in fact ample: that future computers, then in the design stage, would get along on a dozen instruction types, and this was known to be adequate for expressing all of mathematics. (Parenthetically, it is as true today as it was then that "programming" a problem means giving it a mathematical formulation. Source languages which use "plain English" or other appealing vocabularies are only mnemonic disguises for mathematics.) Von Neumann went on to say that one need not be surprised at this small number, since about 1,000 words were known to be adequate for most situations of real life, and mathematics was only a small part of life, and a very simple part at that. This caused some hilarity in the audience, which provoked von Neumann to say: "If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is."