Since 1956 a folder of penciled notes has occupied the rear of whatever filing device I have had in use. Labeled simply "Teaching Staff," the folder contains the hurried scrawl of an untrained secretary—trained neither for rapid transcription nor for efficient selection of detail—a secretary who was selected, not because of expertise in the subjects under discussion, but in keeping with standardized conceptions of what a secretary should look like. The notes are therefore undoubtedly incomplete and may in a few spots be in error. They are, however, the only still-existing record of these early "brainstorming" sessions—unprejudiced by the hindsights of what we wished we might have thought about. It is the fate of historians to work with such records.

These yellowed notes had been left unread for years, having served their purpose in producing the first of the sets of "rules" for programming that have sprung from that primal source of truth—Harvard University. I have assumed in the years since, as would most members of our culture, that programming has

---

1 This paper was read at the convention of the National Society for Programmed Instruction, San Antonio, Texas, April 2, 1964.

The author is assistant professor, School of Education, University of California, Los Angeles.
made tremendous progress and that nothing but benign amusement at the childhood of the art could be forthcoming from a trip backward in time. As a former historian, I should have known better. In that discipline, “progress” is an unfashionable explanation of differences between eras. As a psychologist, I should have known better. In that discipline, observation supports the generalization that rigidity of thought, the hardening of the mental arteries, is a property, not of youth, but of age.

It has become fashionable of late—as late as yesterday, for instance—to attack the so-called formulae of linear programming as archaic birdseed, unfit for human consumption. We have heard complaints from the granddaddy of us all about this modern generation—with its emphasis on the shiny chrome of reinforcement and the push-button ease of errorless learning. The blame, of course, is laid squarely at the feet of the felt proximity of the pigeon lab to the teaching lab on the Harvard campus. My purpose today, in digging up the past, is to destroy, if possible, the myth that what we now know as linear programming sprang full-grown from a cumulative record. It was, rather, shaped by forces not all of which were present in that first year. I leave it for future historians to identify the forces.

Those who attended the fairly regular meetings were B. F. Skinner, project director; Lloyd Homme, Susan Meyer, Douglas Porter, and Irving Saltzman, “programmers”; and two at-that-time graduate students, Wells Hively and Matthew Israel. Observers sat in occasionally. The source of the comments was rarely identified in the notes. The content of the meetings was extensive discussion of the subject matter of several programs under construction in physics, psychology, French, and arithmetic, as well as general principles of programming and problems of machine design. The famous Harvard machine, which in another paper (1) I have called “Tyrannosaurus I,” was already in production in the Harvard shop. One program—Ferster and Sapon’s German vocabulary program—was already in existence. Such, then, was the setting, not of the first signs of programming activity, but rather of the first large-scale effort in the field. In presenting this material, I have translated outlines into sentences, but have added nothing and have attempted to leave the vocabulary unchanged.

The first set of notes is dated 23 September 1956. The various combinations of input and output in a French program were dis-