ideas already known. Thus our educational system in computing should do three things:

1. teach the leading ideas now believed to be relevant to computing, and inspire students with the desire to keep on learning after they graduate;
2. seek out and inspire a few leading minds capable of augmenting our stock of ideas with good new ones and show them how to do it;
3. inspire a generation of students to design and experiment with good systems with the methods now known and soon to be discovered, and show these students how to do a good job of it.

By accepted custom, the Ph.D. degree requires the student to perform in steps (1) and (2): he passes examinations in relevant ideas, and he writes a dissertation whose main requirement is to contain some original theoretical work. I believe these two steps are entirely correct for a Ph.D. in a theoretical subject like pure mathematics.

But the very point of founding schools of engineering and departments of computer science was that society needs concentration on work relevant to today's technology. This implies a certain abandonment of learning for learning's sake, in favor of work on problems whose solution is actually needed. In the computing field, this implies to me that we must not confine our students to Ph.D. dissertations that are of the classical type, but should be prepared also to accept first class work in design or experimentation.

Students are attracted to computer science because it has a lot of action, rather than just contemplation. From the start, our students are creating programs that do things, and they enjoy it. Many of them are eager to keep on designing and programming systems, and it seems almost criminal to turn this eagerness off. Instead, our graduate departments should be accepting this urge to produce, and concentrating on channeling the design, experimentation, and production into worthwhile projects, done with high standards.

It has been argued that design and experimental work are fine, but should be rewarded with a different degree—one analogous to the degrees of Engineer or Doctor of Arts. I disagree, mainly because different degrees tend to acquire different levels in the hierarchy of snobbery, and I refuse to admit that excellent work in design is any less important than excellent work in theory. The Ph.D. degree has become the accepted reward for first class performance in graduate school (e.g., in experimental physics), and should be retained in that function. Any further assessment of the quality of a person's work can be passed along in personal letters of recommendation. If my recommendations were followed, I would expect to see more Ph.D. theses with titles like

"A very high-performance compiler for PL/1 on System/360";

"Study of all calls on the scientific subroutines on the CDC 6600 at NYU in October 1969, and a resulting proposal for reorganizing the library."

In summary, the purpose of our educational establishment is to reward students for developing their educational and performance potential as much as they can. Let us use the Ph.D. as a reward for first class work in any aspect of our field, and not discriminate against work in design or experimentation.

Applied computer science

by LOTFI ZADEH

University of California
Berkeley, California

It is a truism that we are in the throes of an information revolution of which one obvious manifestation is a very rapid growth in the number of users of computers and computer-like information processing systems.

It is also evident that the number of computer users is growing much more rapidly than the number of computer scientists and engineers. As a result, computer science may become user-influenced to a much greater extent than other fields of science and engineering. This was in evidence at the 1968 IFIP Congress in Edinburgh, at which the ratio of non-professional users to computer scientists was far greater than, say, the ratio of non-mathematicians to mathematicians at the 1966 International Mathematical Congress in Moscow.

The overwhelming preponderance of computer users over computer specialists is certain to have a profound impact on computer science education in the years ahead. One likely effect is that much of the training in the use of computers will be taking place outside of computer science departments and will be tailored to the needs of students in particular fields. Another possible effect is a splitting of computer science into pure computer science and "applied" computer science a la the division of mathematics into its pure and applied branches.

If one believes, as this panelist does, that an organiza-