R. N. Caffin’s “Heresy on High-Level Languages” brought forth a flood of letters—three to be exact. I was not surprised to receive one from an APL fan, touting the superiority of his favorite language and ignoring all the issues raised by Dr. Caffin. (Am I the only one in this world who neither adores nor detests APL? It certainly seems to be a polarizing topic.) John Lowry wrote regarding the “C” language, which is the base language of Bell Labs’ Unix system. He notes that C has a certain amount of overhead, such as variable declarations and a procedure entry for the main program, but that it is on the whole a terse language requiring few keystrokes. It has other virtues that we won’t go into here.

The most philosophical letter came from David Feign, and is reproduced here:

Dear Jim,

The Open Channel in the March 1979 Computer makes a point I agree with thoroughly. Modern programming languages are designed to impose discipline on the programmer and to force him to think in certain standard ways. But most people don’t naturally think the way computers and computer scientists would like them to. Forcing a programmer to use certain tools will not automatically straighten out his thinking. We have all seen horrible examples of “structured” programs and good examples of Basic, Fortran, and even assembly-language programs.

The problem is that computer scientists want to make people think and express themselves in a way that is easy to translate into machine code. The much harder problem of understanding how people really think and express themselves, and translating this into a machine language, has been dropped by the computer scientists. Solving the harder problem would mean more work and many “programming” languages rather than the “universal” programming language that they are looking for. If you want people to think and express themselves your way, you have to start training them in kindergarten.

A reason for the popularity of Fortran was that it was not a “programming language” but a system for translating an established language (algebra) into a machine-acceptable language. A few “control” statements were added to the algebra, and people could now write computer programs with only a few concepts to learn.

I learned this lesson years ago. Being a “computer expert,” I knew all about how to use computers. But I was frustrated in my attempts to teach a group of experts in another field how to use this tool—until I learned their language and how they thought about problems and procedures.

When I developed a translator from their language into computer procedures, then my system became acceptable and they used “my” computer. There’s a whole field, called “Human Engineering,” based on this idea.

Make things easy for the user and respond with new ideas only when he’s stuck and needs them. Otherwise, make sure that you know the right way to think, and train infants that way. Only then will computer use be easy and “natural.”

Sincerely,
David Feign
Chapman College

We have just become aware of a new development in programming languages which renders the whole question academic. The following announcement should end all arguments as to whether my language is worse than yours:

**National Semi offers SMUT**

Created to fit the way a system designer naturally thinks, a new, highly evolved, low-level computer language was recently introduced to the world by National Semiconductor Corp. at its headquarters in the heart of Silicon Gulch.

Fondling his company’s newest MKV hand-held computer, Dr. Philo T. Breakthrough explained that the name of the new language is SMUT, which stands for “Simple, Manual, Understandable, and Tacky.” When run on the company’s Emergency Back-Up Hand-Held Digital Computer (EBUHDC), SMUT operates in the user’s native language, which has been found, according to Dr. Breakthrough, to be invariably a base-5. “We wanna say that a language THAT base is 250% times more better and a lot greater than the binary stuff our competitors are using,” he claimed.

Co-developer of SMUT, Messrs. Poisson D’Avril, explained that the language is simply a new adaptation of some rather old concepts, but the way in which it is implemented Nationally is expected to put his firm out in front of everyone concerned with natural law and order. “All other firms are creating higher-level languages in a leap-frog fashion,” D’Avril said. “Eventually, all that activity will come full-circle, and they’ll end up below where they started. Rather than out-race them, we decided to back-track and put our kinky minds
where the rest will eventually arrive— right at the bottom.”

Marketing manager of SMUT, Mitchell Brothers, said that the language “establishes a new low for the industry. Nothing is lower than our SMUT, and we can prove it. We’ve done a lot of work in this field, and we’ve developed a network of service centers in almost all cities with a population greater than 100. And I wanna add that SMUT is immediately available, off-the-shelf, and in several formats, including print, video tape cassette, film, and flash cards. Of course,” he added, “you must be over 18 to buy it.”

On National’s MKV, SMUT does bidirectional calculations equally fast. Optional memory strapping is available, using red, white, and blue ribbons, and an advanced version is chained. Language functions include PUSH, PULL, POP, POKE, STROKE, ADD INTEGER, SUBTRACT INTEGER, and NAIL 1, 2, 3… 5, with a special diagnostic instruction that starts with CLEAN NAILS. Bubble memories can be accommodated with the use of neoprene spheroids. Charge-coupling can be achieved with nylon shag carpets. The start-up diagnostic begins with the command WIGGLE FINGERS.

September 1979

J. H.