time spans. The navigational section contains short representations of the Greenwich Hour Angles (GHA) and declinations of the Sun, Moon, and navigational planets and the GHA of Aries. These functions are represented to navigational precision (±0′.1) as power series of fifth degree, which can easily be evaluated with programmable pocket calculators.

An applications section contains many formulas and algorithms of general utility in navigation and astronomy.

In the astronomical section, Chebyshev series represent such data as the apparent right ascensions and declinations of the Sun, Moon, and planets, nutation in longitude and latitude, sidereal time, and the equation of the equinoxes. Since a small computer is required to evaluate the series efficiently, these data are available in machine readable form. An algorithm for evaluating Chebyshev series is provided in the almanac.

Finally, a list of 176 stars is included. In addition to mean places at the beginning of the year, this list provides data for simply calculating apparent places during the year to a precision of ±0′.1.

The 1978 edition of the *Almanac for Computers* can be obtained by sending a check for $3.00 payable to the U.S. Naval Observatory; 34th and Massachusetts Ave., NW; Washington, DC 20390.

Tip for E.E. Pac Owners

If you have our *E.E. Pac 1,* you'll enjoy—and make use of—this neat improvement to one of the programs.

Dear Sirs:

HP-67 Users who have utilized EE1-18A, "Bilateral Design: Gain and Stability Circles, Load and Source Mapping," may be interested in the following helpful tip.

Besides computing constant gain output circles, the program can readily compute input or constant transducer gain circles. Simply interchanging only the S11 and S22 data entry on card EE-17A, while not altering the data for S21 and S12, allows this computation. No alteration of the program is necessary. This modification of data-input entry extends the usefulness of the program to optimization of transducer gain and noise figure.

To use this tip, load card EE1-17A, then enter S11 data; angle, magnitude, 22. Next, enter S22 data; angle, magnitude, 11. S12 and S21 are left unchanged. Now, load card EE1-18A and compute *input* gain circles.

Sincerely,

Alan Victor, Cooper City, Florida

You're Never Too Young...

Do you remember your 8th birthday? If programmable calculators had existed when you were that age, do you think you could have programmed one? Or do you think that 8 years old is too young to start programming? And if you have guessed by now that we are leading up to something, you're right. We are.

Here is a most remarkable account about an 8-year-old youngster in Israel who can program a 223-step game even though he hasn't yet learned to spell all of the words of calculator jargon we so freely toss around. And the program works; we checked it carefully. Also, he did it by himself, which is all the more startling.

Following is an excerpt from a letter his father wrote to our Product Marketing Manager, then the abstract for the program.

Dear Sir:

Our eldest son, Zvi, has become hooked on programming. It's funny, but he took to it like a fish to water and spends almost all of his free time working out programs for all sorts of things and, for his age (8), really seems to do very well. He's been dying to write a program good enough for the Library, and it is hard for me to explain that it would not be taken. I do want to encourage him, so I promised to send this program (all completely his own work) to HP. It would be super if you could take the trouble to just drop him a note from HP

Sincerely yours,

David Schreiber, Jerusalem

67/97 Noughts-and-Crosses (01884D)*

The game of noughts-and-crosses *(also called tic-tac-toe, Ed.)* is played by two players; one writes noughts (zeros), the other writes crosses (x's) in any of the nine squares in the familiar pattern: #. Players 1 and 2 take turns and "write" a nought or a cross by using the $\boxed{1}$ through $\boxed{9}$ keys on the keyboard. The HP-67/97 will record the move internally but will not show the move in the display. Thus, the other player cannot see which square was chosen. And if a player tries to play twice in a row, the calculator will catch the double move and flash a 0.000 display. If a player tries to put a cross or nought in a square that already contains one, the number (of that square) flashes in the display. When one player gets three noughts (or crosses) in a row, the calculator indicates which player won and which squares were used.

(Congratulations, Zvi, for a sensational job and for proving what all of our Users already know: HP calculators are easy to program. Ed.)

Author: **Zvi Schreiber**
Jerusalem, Israel

In European areas, order by number 00296D.

What Is A "Quality" Program?

Why are *some* programs highlighted in KEY NOTES? Why do *some* programs you buy seem (ahem) "less perfect" than others? Well, after seeing over 7,000 programs in the last 4 years, we can make a few observations.

Most really "good" or "excellent" programs get that attribute because the author has taken the time to *carefully* document the problem, the solution, and *how* it was derived. Examples, for example, make all the difference in the world. Also, just because there are only five sheets in a program submittal package does not automatically limit you to only five sheets. Don't attempt to cram a complex matrix algebra program on one sheet—or five sheets—when six or eight are necessary. And keep in mind that some people have HP-97's (or HP-67's) and program accordingly.

Try to realize that someone else will have to interpret your program, perhaps someone who doesn't know the subject matter as well as you do. So, when you find an obscure solution to a tricky problem, reference it! Take the time to show how you got from step A to step B. Don't be afraid of *too much* detail; it can be invaluable to the neophyte and is easily "skimmed over" by the professional.

When necessary, list the definition of symbols used. Show the derivation of equations and algorithms. Flow charts and diagrams are invaluable to *your* compilation of the program; don't short-change the final submittal by leaving them out. Also, a concise and usable sample problem(s) usually will get most people acquainted with the subject so they will be able to *apply* the program. And, last but not least, make the program listing totally useful by carefully listing all register contents and completely describing all labels.

Remember, when *everyone* submits well-documented, high-quality programs, all of *you* are the beneficiaries, not the Library.