

The Monitor has some special features that make it easier for you to write and debug machine-language programs, but before you get into that, here is a small machine-language program that you can run using only the simple Monitor commands already described. The program in the example merely displays the letters *A* through *Z*: you store it starting at location \$0300, examine it to be sure you typed it correctly, then type 300G to start it running.

```
*300:A9 C1 20 ED FD 18 69 1 C9 DB D0 F6 60
```

```
*300.30C
```

```
0300- A9 C1 20 ED FD 18 69 01
```

```
0308- C9 DB D0 F6 60
```

```
*300G ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
*
```

Disassembled programs

Machine-language code in hexadecimal isn't the easiest thing in the world to read and understand. To make this job a little easier, machine-language programs are usually written in assembly language and converted into machine-language code by programs called **assemblers**.

Since programs that translate assembly language into machine language are called assemblers, a program like the Monitor's LIST command that translates machine language into assembly language is called a **disassembler**.

The Monitor's LIST command displays machine-language code in assembly-language form. Instead of unformatted hexadecimal gibberish, the LIST command displays each instruction on a separate line, with a three-letter instruction name, or **mnemonic**, and a formatted hexadecimal operand. The LIST command also converts the relative addresses used in branch instructions to absolute addresses.

The word **mnemonic** comes from the same root as *memory* and refers to short acronyms that are easier to remember than the hexadecimal operation codes themselves: for example, for *clear carry* you write CLC instead of \$18.