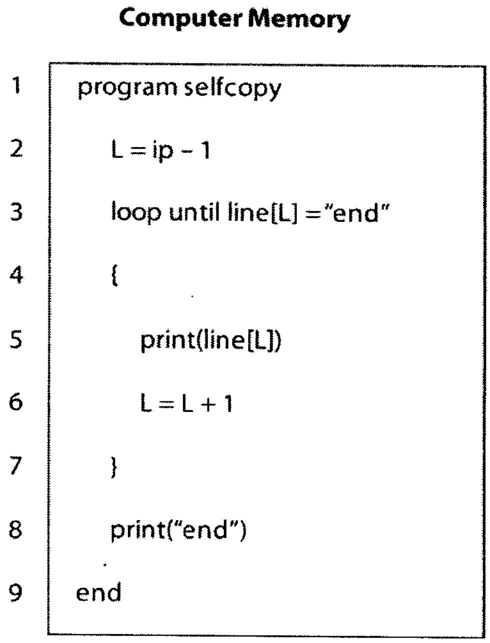
FIGURE 8.2. A self-copying program.



```
1  program selfcopy
2     L = ip - 1
3     loop until line[L] ="end"
4     {
5        print(line[L])
6        L = L + 1
7     }
8     print("end")
9  end
```

operating system—sets the instruction pointer to 1, which points to the name of the program. The ip then moves down, line by line, executing each instruction.

In memory location 2 a variable L is set to ip−1. Recall that ip is the location of the instruction currently being executed. So when line 2 is executed, ip is set to 2 and L is set to 2 − 1 = 1. (Note that L will now stay equal to 1 until it is reset, even though ip changes as each instruction is executed.)

Next, a loop is entered, which will be iterated until line[L] is equal to the character string end. Remember that line[L] is equal to the string located in memory location L. Right now, L is set to 1, so line[L] is equal to the string program selfcopy. This is not equal to the string end, so the loop is continued. In the loop, line[L] is printed and L is incremented. First, with L = 1, program selfcopy is printed; then L is set to 2.

Now, line[L] is the second line of the program, namely L = ip - 1. Again, this string is not equal to end, so the loop is continued. In this way, each line of the program is printed out. A particularly interesting line is line 5: when line 5 is being executed with L = 5, the instruction print(line[L]) prints itself out. When L = 9 and line[L] is equal to end, the loop ends. At this point, lines 1–8 have been printed. The instruction pointer moves