

## In-lab 6 Preparation

To prepare for In-lab Exercise 5, get the finished version of the `ReadLine` function source code from the web site for the course, with the name `ReadLine.a68`, and read it over. It uses the stack to save registers it will use so that only the output register `d0` is altered (we'll talk about this in a lecture soon).

It is used by placing the address of a buffer into `a0` and the buffer's size into `d0`. `ReadLine` will return the number of bytes read in `d0`, or place `-1` there to indicate that end-of-file has been detected, and leave `a0` unchanged. It will use `Ctrl-D` (`0x04`) for end-of-file, `ENTER` (`0x0D`) for end-of-line, `Backspace` (`0x08`) for (what else?) backspacing, and `Ctrl-X` (`0x18`) to erase the current input line. All other control characters (the range from `0x00` to `0x1F` inclusive) are ignored and are not put into your buffer. The `CR` character is replaced by the `NUL` (one-ell!) end-of-string character (`0x00`).

You can assemble `ReadLine.a68` but do not try to link it yet.

You may start to design a program to test `ReadLine` by writing your PDL and Data Dictionary. Be sure to create a `.lcf` file for your test program, but look at this lab for the details.