Maiah Pardo
CruzID: mapardo

Write Up: asgn0

- **Testing on shoulders.c**
  - Tested with no arguments
  - Tested with one argument – a number / an invalid number / a text file as only argument
  - Tested with randomely placed dashes (-)
  - Tested with large and small text files, large and small binary files, and a mix of these and dashes
  - Tested with a large number of lines
  - Used the 'diff' command to check for differences between the output of head and my program
  - Tested for memory leaks using valgrind (not sure this was entirely necessary)

- **How does the code for handling a file differ from that for handling standard input? What concept is this an example of?**
  - When handling a file you must open the file and check that there was not an error in opening the file (if the file descriptor is equal to -1, there was an error in opening the file). Then when reading using the read() function, we must give the file descriptor as the first argument.
  - Handling standard input differs greatly from handling a file, as we do not have to open or close anything. All we do is pass the posix name "STDIN_FILENO" (which is equal to 0) to the read function which represents reading from standard in.
  - This is an example of how memory stores different types of inputs in different components of our computer's hardware. Memory is one of the fundamental abstractions (as discussed in section 2.1 of the reading). The first time we ask to read from a file (which is already saved on the computer), the file lies in the hard drive and must be read and sent to RAM, while standard input is just stored in RAM.