

## Design Document: Asgn 0

### Program: Shoulders

- The goal of this program is to implement a limited version of the “head” program in quiet mode and line number mode, without support for any flags. If a “-” is given as a file, shoulders uses standard input. If no file names are given, standard input is echoed to standard output for the certain number of lines. Each newline character is considered the end of a line. If there is an error reading the file, an error message is printed to standard error and the next file will be read.
- The program must be written in C, and may not use any of the C library FILE \* functions such as fread() and printf() for user data. We also may not use fopen(). We may not use functions that allow us to execute external programs, such as system(3). We may use fprintf() (in one case only), perror(), and warn() for error messages. We may also use sprintf(). We must use fixed-sized buffers. The memory will be limited to 10 MB during execution.

### Functions:

- **int main(int argc, const char \* argv[])**
  - o Takes in arguments which consist of a number representing number of lines to show and then an optional any number of file names or a dash for standard input. If no file names are given, the program will take standard input for the desired number of lines.
  - o main() parses through all of the given arguments:
    - determines if the first is a number
      - if it is, continue
      - if it is not, quit with error message
    - determines if the rest are filenames or dash
      - if it is a filename, opens the file and calls io() function
      - if it is a dash, calls echo\_input()
    - if there are no file names
      - calls echo\_input()
- **void io(int max\_lines, char\* Buffer, int bufferSize, int fd)**
  - o Takes in an integer representing the number of lines to show, a Buffer, buffer size, and file descriptor.
  - o Each newline read in is counted as one line of the file, and lines are printed until the max number of lines is reached
    - Buffer passed in is of constant size 256 (used for reading)
    - buff (another buffer declared inside) is of constant size 16 (used for writing)
    - checks for newlines and writes the corresponding number of lines

- **void echo\_input(int max\_lines, char\* Buffer, int bufferSize, int fd)**
  - Takes in an integer representing the number of lines to echo, a Buffer, buffer size, and file descriptor (the file descriptor is used for reading standard in)
  - echo\_input() reads standard input for the given number of lines, and outputs them to standard out.
    - Buffer passed in is of constant size 256 (used for reading)
    - buff (another buffer declared inside) is of constant size 1 (used for writing)
    - writes standard input to standard output until the desired number of lines is reached