Jack Tomkiel
CSC 345-01 – Project #1: User Interface
02/18/22

For this project, all requirements were fully implemented. Below is a list of each of the requirements, as well as some details that demonstrate the degree of their functionality.

- **[10 pts]** Child process creation and executing commands
    - Fully implemented
    - Child process is forked and executes commands given by the user, using execvp as described in the project description and textbook. Additionally, another child process is forked for the pipe process. If & is included, the processes execute concurrently, if not, the parent process waits for the child process to finish.

- **[10 pts]** Command history management
    - Fully implemented
    - If the user enters !!, the previous command is echoed, executed again, and placed in the history buffer as the next command.

- **[10 pts]** Add support of input and output redirection
    - Fully implemented
    - Shell supports '>' and '<' redirection operators. The output command redirects the output of a command to a file and the input command redirects the input to a command from a file as expected. The dup2() function is used and both test cases described in project hand out work correctly.

- **[10 pts]** Allow the parent and child processes to communicate via a pipe
    - Fully implemented
    - Communication via a pipe allows for the output of one command to serve as the input of another. Both commands run as separate processes and function as expected. The test case described works correctly, and other test cases such as ls | less, with differing numbers of arguments are also handled.

- **[5 pts]** Keep working directory information (change directory by [cd] command)
    - Fully implemented
    - [cd] command was created and allows the user to change directories the same way as a typical shell would.

- **[5 pts]** Up-to-date display the current directory in the prompt. For example, if you are in **/home/user/my**, then your prompt should show as
  osc:/home/user/my> instead of osc>
    - Fully implemented
    - Also used ANSI escape codes to print bold & color text as shown above