

## Assumptions:

- The terminal prompts the user with new input by printing current directory of user with the ">" sign, this is not the normal behaviour on bash since actual bash uses \$ instead of >
- Terminal exits when the user inputs an empty input.
- For cat command, when we have to read from both standard input and specified file, it will consider the contents of the file first and then the contents from standard input while outputting. This is specifically for the catFilesAndStdin method and not the run method (read point below).
- The description of cat in the assignment states that "The cat command concatenates the content of given files" and that "With no FILES, or when FILES is -, read standard input." Hence this would mean that standard input is read only when there are no files, and that when files are present there is no need to read standard input.
- If multiple same files are presented as arguments to a method like tee, which has the option of writing/appending to multiple files, the content will be inserted multiple times (assuming the append flag is true to the same file)
- For the uniq command, the "-D" option will take precedence over the "-c" option. So this means that when both options are present together, the output will print all duplicated lines.
- In the cut command, an error should be thrown if both character and byte position flags are specified or absent simultaneously.
- When determining character positions, the newline character (\n) is counted as a character
- If the provided range does not correspond to a valid substring of the input, cut will output an empty line for that input line
- In the cut command, if the starting value of the range is less than 1 or if the ending value is less than the starting value, it is considered an invalid range. In such cases, an exception should be thrown to indicate the invalid range specification for the cut operation
- Sequence command when used with only one command should throw an exception as there is no follow up command
- There were a lot of TDD cases given from Prof that do not work on windows and are not failing due to our implementation (Hence, they are disabled when running on window os).

- There are no trailing tab spaces in the last line of paste output. This could happen when first file has more lines than the second file, so in that case there will not be a tab between file1 content and non-existent line of file2.
- If there is any problem with one of the files entered, the paste command stops executing and throws an exception. This is because continuing will give the user a false idea that the file entered by them has been processed in the output.
- For the mv command, if the user does not have write permissions for the source file, but has write permissions for the destination directory, the mv command will move the file to the destination directory.
- For any issue related to file not found or directory is empty (or other directory related issues) or no permission to a file, we choose to throw an exception rather than return an error message
- For cat, uniq and sort commands, we have a new line at the end that is created in the run method of the command as opposed to the individual file or stdin (or any other) methods defined in the command interface. However for cut, we choose to add new line in the individual methods separately rather than the run method
- If only '-' is specified as part of files in the argument list, we tend to call the stdin method since there are no files
- However, if '-' is specified as part of a longer list of files, we choose to read stdin and files. This approach relies on the user to use the "-" argument to explicitly state when stdin should be read, which is a common convention in Unix-like systems for tools that operate on both files and standard input
- For cat, we assume that the order of concatenation is not of particular importance when we read from both file and stdin. First we append file contents followed by stdin contents if the catFileAndStdin method is executed
- Similarly, for catFileAndStdin, we assume that the line numbers are continuous. This means that when stdin contents are appending to the file contents in a single file, the line number of the stdin contents will be a continuation of last line number of the file contents as opposed to starting from 1
- For sort, we consider '-' as an invalid file name and throw an exception since there is no mention of it on the document. For other commands such as Cat and Cut, '-' is ignored if it is part of a large file list (an exception is not thrown)
- For ls, when calling 'ls -R' the first line does not have a './:' as header for the contents in that current working directory.
- For grep, when calling 'grep -H', output will be <fileName>:<lineContent> with **no space** between ':' and <lineContent>. (e.g. 'grep -H "a" README.md', output is 'README.md:# CS4218codebase')