Final Assignment

Question 1

- Awk
    - Description
        - Awk is a scripting language used for processing and displaying text
    - Formula/Syntax
        - awk + options + {awk command} + file
    - 3 Examples that we understand well
        - awk '{print $1}' cars.csv
            - will return the first column of every line in the file, similar to cut
        - awk -F {print $1," = ", $NF} /etc/passwd
            - will display the first and very last column of the file
        - awk -F {print NR,$1,$3}
            - will display the first 3 fields with line numbers
- cat
    - Description
        - Cat is used for displaying the content of a file
    - Formula/Syntax
        - cat + option + file(s) to display
    - 3 Examples that we understand well
        - cat -n todo.md
            - display with line numbers
        - cat -b todo.md
            - display content with line numbers excluding empty lines
        - cat -s todo.md
            - multiple empty lines are turned into one empty line
- cp
    - Description
        - copies files and directories from a source to a destination
    - Formula/Syntax
        - cp + files to copy + destination
    - 3 Examples that we understand well
        - cp Downloads/wallpapers.zip Pictures/
            - will move the wallpapers.zip into Pictures/
        - cp ./* ../
            - copy the files from the entire directory onto the file above
        - cp -r ~/Downloads/wallpapers ~/Picture/
            - copy a directory
- cut

    - Description
        - extract specific section of eac line of a file to display on a screen
        - These types of files must be separated by some kind of separator character. One of the options will be which character to look for as a separator
    - Formula/Syntax
        - cut = option + file(s)
    - 3 Examples that we understand well
        - cut -d ':' -f1 /etc/passwd
            - this will display the first 'column' of the file
        - cut -d ':' -f1,7 /etc/passwd
            - this will display the first column and the last column side by side
        - cut -d ':' -f1,7 --output-delimiter=' => ' /etc/passwd/
            - This will change the separator ':' to '=>'

- grep

    - Description
        - Search text in a given file. grep works line by line basis
    - Formula/Syntax
        - grep + option + 'search criteria' + file(s)
    - 3 Examples that we understand well
        - grep 'dracula' dracula.txt
            - grep 'dracula' dracula.txt
                - will look for any line that contains the word 'dracula'
            - grep -i 'dracula' dracula.txt
                - will disable case insensitivity it will match the letter regardless
            - grep -c 'dracula' dracula.txt
                - will display the number of times that word is found in the file

- head

    - Description
        - displays the top N number of lines of a given file
    - Formula/Syntax
        - head + options + file(s)
    - 3 Examples that we understand well
        - head ~/Documents/Book/dracula.txt
            - Display first 10 lines of a file
        - head -5 ~/Documents/Book/dracula.txt
            - Displays the first 5 lines of a file
        - head -3 ~/Documents/Book/dracula.txt
            - Displays the first 3 lines of a file

- ls

    - Description
        - used for displaying all the files inside a given directory
    - Formula/Syntax

        - ls + directory
    - 3 Examples that we understand well
        - ls -a
            - list all the files inside the directory including hidden files
        - list ~/Pictures
            - list all the files inside the absolute path directory
        - ls -S
            - list all the files by Document size inside the current working directory

- man
    - Description
        - Man is used to give a list of options and explain the command
    - Formula/Syntax
        - man + command
    - 3 Examples that we understand well
        - man ls
            - gives an explination of ls and a list of options
        - man cd
            - gives an explination of cs and a list of options
        - man tree
            - gives an explination of tree and a list of options

- mkdir
    - Description
        - used for creating directories
    - Formula/Syntax
        - mkdir + name of directory
    - 3 Examples that we understand well
        - mkdir wallpapers
            - will create a wallpaper directory
        - mkdir wallpapers/ocean
            - create a directory within a directory
        - mkdir ~/wallpapers/forest
            - create a directory using absolute path

- mv
    - Description
        - moves and renames directories or files
    - Formula/Syntax
        - mv + source + destination
    - 3 Examples that we understand well
        - mv Downloads/homework.pdf Document/
            - move a file using relative path
        - mv touch.txt touch2.txt
            - rename the file touch to touch 2

        - mv Downloads/english_homeowork.docx /media/student/flashdrive/
            - moves the file from one directory to another

- tac
    - Description
        - displays contents of a file in reverse order
    - Formula/Syntax
        - tac + option + files(d) to display
    - 3 Examples that we understand well
        - tac todo.md
            - display the content of todo.md backwards
        - tac ~/DOcuments/toto.md
            - displays the content of todo.md backwards using absolute path
        - tac python.py
            - displays content of the file backwards

- touch
    - Description
        - creating files
    - Formula/Syntax
        - touch + file name
    - 3 Examples that we understand well
        - touch list
            - create a file calles list
        - touch list_of_cars.txt script.py
            - creates two files
        - touch ~/Downloads/games.txt
            - creates a file withing a directory using absolute path

- tail
    - Description
        - the tac command is used for displaying the content of a file in reverse order
    - Formula/Syntax
        - tac + option + file(s) to display
    - 3 Examples that we understand well
        - tail dracula.txt bible.txt
            - displays last 10 lines from each txt file but they have headers separating each file
        - tail dracula.txt
            - displays the last 10 lines of a file
        - tail -5 dracula.txt
            - displays last 5 lines of a file

- tr
    - Description
        - the tr command is used for translating or deleting characters from standard output

    - Formula/Syntax
        - standard output | tr + option + set + set
    - 3 Examples that we understand well
        - cat file.txt | tr '.' ':'
            - This replaces the '.' character into a ':'
        - cat program.py | tr '/etc/passwd/' ' '
            - Replaces the directory in the output to a blank space
        - cat file.py | ":space:' '\t'

- tree

    - Description
        - list contents of a directory in a tree like format
    - Formula/Syntax
        - tree + directory to list
    - 3 Examples that we understand well
        - tree
            - will list all the files in the current working directory
        - tree ~/Documents
            - will tree list all the files in Documents
        - tree ~/Downloads
            - will tree list all the files in Downloads

Question 2

- How do you work with multiple terminals open
    - by right clicking on the terminal you can select new window and split screen to create two terminals
- How to work with manual pages
    - man + command
    - press h for help with the man page
    - press q to leave the manual page
- How to parse for specific words in the manual page
    - using /pattern you can search for specific words withing the manual
- How to redirect output (> and |)
    - using the greater than symbol will write the output of a command to a textfile
    - using two of them will keep the old data and write onto it
    - using the pipe character will redirect the output of one command into another command
- How to append the output of a command to a file
    - using two greater than symbols we can append the output of a command to a file
- How to use wildcards for copying and moving multiple files at a time
    - Wildcards are used for selecting many files containing a certain collection of characters
    -
        - will match anything and nothing, with any number of characters
        - ls *.txt
            - this command will return every file with text in front of a .txt file
    - ? will match exactly one character
        - ls b??k

            - this will return every file name where there are two characters withing b and k
        - bracket wildcard will match a single character within a rance, basically regular expressions
            - ls f/aeiou/
                - this will return any file that contains a vowel after the letter f
- How to use brace expansion for creating entire directory structures in a single command
    - Brace expansion will let you select multiple of one file if they match a name
    - mkdir -p music/{jazz,rock}/{mp3files,videos,oggfiles}
        - this command will create multiple directories within the music directory
            - jazz and rock
                - and then withing jazz and rock BOTH will have
                    - mp3files,videos,oggfiles