Final Assignment

Question 1

For every command in this list, include a description, the formula/syntax, and 3 examples that you understand well.

- awk
    - a scripting language used for processing and displaying text; can work with a text file or standard output
    - Formula: `awk + options + {awk command} + file + file to save (optional)`
    - Examples:
        - print the first column of every line of a file
            - `awk '{print $1}' ~/Documents/Csv/cars.csv`
        - print the last field of the /etc/passwd/ file
            - `awk -F: '{print $NF}' /etc/passwd`
        - converts the first field to lower case
            - `awk -F: '{print tolower($1)}' /etc/passwd`
- cat
    - Used for displaying the content of a file, short for concatenate
    - Formula: `cat + option + file(s)`
    - Examples:
        - displays the content of a file with line numbers
            - `cat -n ~/Documents/todo.md`
        - displays the content of a file in your current directory
            - `cat todo.lst`
        - displays the content of a file suppressing repeating empty lines to a single empty line
            - `cat -s ~/Documents/todo.md`
- cp
    - copies files/directories from a source to a destination
    - Formula: `cp + files to copy + destination`
    - Examples:
        - to copy a file
            - `cp Downloads/louvre.png Pictures/`
        - to copy a directory with absolute path
            - `cp -r ~/Downloads/roadtrip2023 ~/Pictures/`
        - to copy the content of a directory to another directory
            - `cp Downloads/roadtrip2023/* ~/Pictures/`
- cut
    - used to extract a specific section of each line of a file and display it
    - Formula: `cut + option + file(s)`
    - Examples:
        - displays a list of all users in your system
            - `cut -d ':' -f1 /etc/passwd`
        - cut a range of bytes per line

            - `cut -b 1-5 usernames.txt`
        - cut a file excluding a given field
            - `cut -d ',' --complement -s -f3 users.txt`
- grep
    - used to search text in the given file, works on a line by line basis
    - Formula: `grep + option + search criteria + file(s)`
    - Examples:
        - search any line that contains the word "shrek" in the given file
            - `grep 'shrek' ~/Documents/shrek.txt`
        - search any line that contains the word "shrek" regardless of the case
            - `grep -i 'shrek' ~/Documents/shrek.txt`
        - search and match only the word
            - `grep -o 'shrek' ~/Documents/shrek.txt`
- head
    - displays the top N number of lines in a given file
    - Formula: `head + option + file(s)`
    - Examples:
        - displays the first 10 lines of a file
            - `head ~/Documents/EN101/Essay5.docx`
        - displays the first 100 lines of a file
            - `head -100 ~/Documents/EN101/Essay5.docx`
        - displays the first 5 lines of a file
            - `head -5 ~/Documents/EN101/Essay5.docx`
- ls
    - used for viewing the content of a directory and details of the files and directories
    - Formula: `ls + directory`
    - Examples:
        - list all files recursively
            - `ls -R Downloads/`
        - list all files in a single column
            - `ls -1 Downloads/`
        - list all jpeg files
            - `ls *.jpg`
- man
    - documentation files that describe commands, executable programs, etc.: quick references
    - Formula: `man + command`
    - Examples:
        - open a specific page for the mkdir command
            - `man 3 mkdir`
        - show all the available pages of a command
            - `man -a ls`
        - opens the man page of the cat command
            - `man cat`
- mkdir
    - used for creating directories
    - Formula: `mkdir + the name of the directory`

  - Examples:
    - to create a directory in your pwd
      - `mkdir photos`
    - to create a directory using absolute path
      - `mkdir ~/photos/roadtrip`
    - to create a directory and it's parent at the same time
      - `mkdir -p videos/springconcert23`
- mv
  - moves and renames directories
  - Formula: `mv + source + destination`
  - Examples:
    - to move a file from one directory to another using relative path
      - `mv Downloads/project5.docx Documents/CIS108`
    - to move a file from one directory to another using absolute path
      - `sudo mv ~/Downloads/wallpaper /usr/share/wallpapers`
    - to rename a file
      - `mv project5.txt project5script.txt`
- tac
  - used to display the content of a file in reverse order
  - Formula: `tac + option + file(s) to display`
  - Examples:
    - display the content of a file using absolute path
      - `tac ~/Documents/todo.md`
    - displays the content of a file with line numbers
      - `tac -n ~/Documents/todo.md`
    - displays the content of a file suppressing repeating empty lines to a single empty line
      - `tac -s ~/Documents/todo.md`
- tail
  - displays the last N number of lines in a given file
  - Formula: `tail + option + file`
  - Examples:
    - displays the last 10 lines of a file
      - `tail ~/Documents/EN101/Essay5.docx`
    - displays the last 100 lines of a file
      - `tail -100 ~/Documents/EN101/Essay5.docx`
    - displays the last 5 lines of a file
      - `tail -5 ~/Documents/EN101/Essay5.docx`
- touch
  - used for creating files
  - Formula: `touch + name of file`
  - Examples:
    - create a file
      - `touch cars.png`
    - create a file using absolute path
      - `touch ~/videos/cars3.mp4`
    - create several files

            - `touch finalessay.docx project5.py finalassignment.md`
- tr
    - used for translating or deleting characters from standard output
    - Formula: `Standard Output | tr + option + set + set`
    - Examples:
        - translate one character to another
            - `cat_file.txt | tr '.' ','`
        - translate white space into tabs
            - `project5.py | tr "[:space:]" '\t'`
        - translate tabs into space
            - `project4.py | tr -s "[:space:]" ' '`
- tree
    - a recursive directory listing program that produces a depth-indented listing of files
    - Formula: `tree + option + file`
    - Examples:
        - list files in current directory
            - `tree`
        - List files with entered pattern
            - `tree -P sample* .`
        - prints the output by last modification time instead of alphabetically
            - `tree -t Downloads/`

Question 2

Answer each question.

- How to work with multiple terminals open?
    - Using one window to show the help or manual pages for a command or looking a the output of your commands and another as the working terminal for all of your commands
- How to work with manual pages?
    - Type man followed by the Linux command name whose man page you want to see. The output is long, so use the mouse scroll wheel or the up and down arrow keys to navigate.
- How to parse (search) for specific words in the manual page
    - Use the grep command to search for specific words.
- How to redirect output (> and |)
    - Use either the ">", or the "|" (pipe) operator which sends the standard output of one command to another command as standard input.
- How to append the output of a command to a file
    - Using the >> symbol, it is defined as the append redirection operator.
- How to use wildcards
    - For `*`, used alone to match anything and nothing and any number of characteristics. Use when you want a list of all files with a particular file extension, do not remember the complete file name, or to copy, move, or removed all files that match a specific naming convention.
    - For `?`, matches precisely one character and proves very useful when working with hidden files.
    - For `[]`, matches a single character in a range; uses the exclamation mark to reverse the match.
- How to use brace expansion

    - Is not a wildcard, but is another feature of bash that allows you to generate arbitrary strings to use with commands.