Question 1

awk

- Description

  - awk is a scripting language used for processing and displaying text. awk can work with a text file or from standard output. awk performs operations line by line.

- Syntax/Formula

  - `awk + options + {awk command} + file`
  - `command output|awk + options + {awk command}`

- Examples:

  - how to print the first and last field of a file
    - `awk -F: '{print $1," = ",$NF}' /etc/passwd`
  - how to start printing from a different line
    - `awk 'NR > 3 { print }' /etc/passwd`
  - how to change a field to upper case:
    - `awk -F:'{print toupper($1)}'`
  - how to print the first field of a file
    - `awk -F: '{print $1}' /etc/passwd`

cat

- Description

  - The cat command is used for displaying content of a file. Also used for concatinating files

- Syntax/Formula

  - `cat + option + file or files to view/concatinate`

- Examples:

  - how to see the content of a file:
    - `cat /etc/passwd`
  - how to see the content of a file with line numbers
    - `cat -n /etc/passwd`
  - how to see the content of a file with ending line character
    - `cat -E /etc/passwd`
  - how to display the content of a file suppressing repeating empty lines to a single empty line.
    - `cat -s /etc/passwd`

cp

- Description

- Copies files/directories from a source to a destination.

- Syntax/Formula

    - cp + file or files to copy + destination

- Examples:

    - How to copy directories
        - `cp -r /Downloads/wallpapers ~/pictures/`
    - How to copy multiple files in a single command
        - `sudo cp -r script.sh program.py home.html assets/ /var/www/html/`
    - How to copy a file
        - `cp Downloads/wallpapers.zip Pictures/`
    - How to copy the content of a directory to another directory
        - `cp Downloads/wallpapers/* ~/pictures/`

cut

- Description

    - The cut command is used to extract a specific section of each line of a file and display it to the screen. Each line has about seven fields delimited by colons (😃

- Syntax/Formula

    - cut + option + file or files

- Examples:

    - How to display a list of all the users in your system
        - `cut -d ':' -f1 /etc/passwd`
    - How to cut a file using a delimiter but changing the delimiter i the output.
        - `cut -d ':' -f1,7 --output-delimiter=' => ' /etc/passwd`
    - How to display a list of all users in your system with their log in shell.
        - `cut -d ':' -f1.7 /etc/passwd`
    - How to cut a file excluding a given field
        - `cut -d ',' --complement -s -f3 users.txt`

grep

- Description

    - grep is used to search text in a given file. Also works line by line basis.

- Syntax/Formula

    - grep + option + search criteria + file or files

- Examples:

    - How to search any line that contains a certain word

- - - `grep 'dracula' ~/Documents/dracula.txt`
  - How to search any line that contains a certain word regardless of the case
    - `grep -i 'dracula' ~/Documents/Books/dracula.txt`
  - How to search and display the total number times a given word appears in a file
    - `grep -wc '/bin/bash' /etc/passwd`
  - How to search for all the lines that start with a capital letter
    - `grep -n '^[A-Z]' ~/Documents/Books/war-and-peace.txt`

head

- Description

  - The head command displays the top N number of lines of a given file. By default it prints the first 10 lines.

- Syntax/Formula

  - head = option + file or files

- Examples:

  - How to display the first 10 lines of a files
    - `head ~/Documents/Book/dracula.txt`
  - How to display the first 5 lines of a file
    - `head -5 ~/Documents/Book/dracula.txt`

ls

- Description

  - ls is used for listing the content of a given directory or the files/directory itself.
  -

- Syntax/Formula

  - ls + option + directory to list

- Examples:

  - how list all the options of the ls command
    - `ls --help`
  - how list all the files inside the current working directory including hidden files
    - `ls -a`
  - how to list all the files in a given directory sorted by file size
    - `ls -s ~/Documents`
  - how to long list all the files inside a given directory recursively
    - `ls -1R ~/Pictures`

man

- Description

- man (manual) pages helps users understand how commands work and can be used for quick references.

- Syntax/Formula

    - man + command
    - to exit man page
    - "q"

- Examples:

    - How to open the man page for a command
        - `man passwd`
    - How to open a specific man page for a command
        - `man 5 passwd`
    - How to show all the available pages of a command
        - `man -a passwd`
    - How to show the man page section of the passwd command

mkdir

- Description

    - mkdir is used for creating a single directory or multiple directories.

- Syntax/Formula

    - mkdir + the name of the directory

- Examples:

    - How to create a directory with a parent directory at the same time
        - `mkdir -p wallpapers_others/movies`
    - How to create multiple directories
        - `mkdir wallpapers/cars wallpapers/cities wallpapers/forest`
    - How to create a directory with space in the name
        - `mkdir wallpapers/new\ cars`
        - `mkdir wallpapers/'cites usa'`
    - How to create directory with a single quote in the name
        - `mkdir wallpapers/"majora's mask"`

mv

- Description
    - mv moves and renames directories.
- Syntax/Formula
    - mv + source + destination
- Examples:
    - How to rename a file
        - `mv homework.docx cis106homework.docx`

- How to move multiple directories/files to a different directory
  - `mv games/ wallpapers/ rockmusic/ /media/student/flashdrive/`
- How to move a file from one directory to another combining absolute path and relative path
  - `mv Downloads/english_homework.docx /media/student/flashdrive/`
- How to move a file from a directory to another using relative path
  - `mv Downloads/homework.pdf Documents/`

tac

- Description

  - The tac command is used for displaying the content of a file in reverse order.

- Syntax/Formula

  - tac + option + files or files

- Examples:

  - How to display the content of a file located in the pwd
    - `tac todo.md`
  - How to display the content of a file using absolute path
    - `tac ~/Documents/todo.md`

tail

- Description

  - The tail command displays the last N number of a given file.

- Syntax/Formula

  - tail + option + file

- Examples:

  - How to display the last 10 lines of a file
    - `tail ~/Documents/Book/dracula.txt`
  - How to display the last 5 lines of a file
    - `tail -5 ~/Documents/Book/dracula.txt`

touch

- Description

  - touch is used for creating files

- Syntax/Formula

  - touch + file

- Examples:

- How to create a file with space in its name
  - `touch "list of foods.txt"`
- How to create a several files
  - `touch list_of_cars.txt script.py names.csv`
- How to create a file
  - `touch list`
- How to create a file using absolute path
  - `touch ~/Downloads/games2.txt`

tr

- Description

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

- Formula

  - standard output | tr +option + set + set

- Examples:

  - How to translate one character to another (For example a period with a comma)
    - `cat file.txt | tr '.' ','`
  - How to translate white space into tabs
    - `cat program.py | tr "[:space:]" '\t'`
  - How to translate tabs into space
    - `cat file.py | tr -s "[:space:]" ' '`

tree

- Description

  - list all files and directories in a given directory in a nice tree format.

- Syntax/Formula

  - tree + command

- Examples:

  - How to display a tree like format for a directory
    - `tree /Downloads`
  - How to display only directory entires in output
    - `tree -d`
  - How to display hidden files
    - `tree -a`

vim/nano

- Description

  - They are command line text editors

- Syntax/Formula

    - vim
    - nano

- Examples:

- How to start a vim with a file name

    - `vim food.txt`

- How to save and exit a vim file

    - `vim food.txt, esc key :wq`

- How to search for text in nano

    - `ctrl+w`

- How to start a nano with a file name

    - `nano food.txt`

Question 2

- How to work with multiple terminals open?
    - open one terminal then another terminal and set them side by side. you can also use tillix and split the terminal if needed.
- How to work with manual pages?
    - It displays the manuel for any of the commands in a terminal.
- How to parse (search) for specific words in the manual page
    - In order to search specific words in the manuel page would be `man -k file`
- How to redirect output (> and |)
    - In order to redirect an output it would be `command output + > file`
- How to append the output of a command to a file
    - would have to add >> at the end of the command
- How to use wildcards (For copying and moving multiple files at the same time)
    - you would have to use the `*`
- How to use brace expansion (For creating entire directory structures in a single command)
    - `mkdir -p music/{jazz,rock}/{mp3files,vidoes,oggfiles}/new{1..3}`