Question 1

awk

- Description:
  - awk is scripting language used for manipulating data and generating reports.
- Formula:
- `awk + options + {awk command} + file`
- `command output | awk + options + {awk command}`
- Examples:
  - how to print the first field of a file:
    - `awk -F':' '{print $1}' /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)}'`

cat

- Description:
  - Used for seeing the 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 endling line character
    - `cat -E /etc/passwd`

cp

- Description:
  - Used for copying files from one location to another.
- Syntax/Formula:
  - `cp + option + source_file destination_file`
- Examples:
  - Copying file to target directory.
    - `cp /etc/passwd /mnt/backup/`
  - Copying files interactively, can only work if the destination directory already has the same file.
    - `cp -i /etc/passwd /mnt/backup/passwd`

    - Copying a directory
        - `cp -r /home/linuxtechi /mnt/backup/`

cut

- Description:
    - Cutting out the sections from each line of files and writing.
- Formula:
    - `cut + option + filename`
- Examples:
    - To extract the specific bytes -b with list of numbers seperated by a comma.
        - `cut -b 1,2,3 state.txt`
    - To cut by character use the -c option.
        - `$cut -c [(k)-(n)/(k),(n)/(n)] state.txt`

grep

- Description:
    - Grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.
- Formula:
    - `grep [options] pattern [files]`
- Examples
    - Enables to search for a string case insensitively in the given file.
        - `grep -i "UNix" newfile.txt`
    - Displaying the count of number of matches.
        - `grep -c "unix" newfile.txt`

head

- Description:
    - Prints the top N number of data of the given input.
- Formula:
    - `head + option + file`
- Examples
    - Prints the first ‘num’ lines instead of first 10 lines
        - `head -n 5 valorant.txt`
    - Prints the first ‘num’ bytes from the file specified.
        - `head -c 6 valorant.txt`
    - By using this option, data from the specified file is always preceded by its file name.
        - `head -v valorant.txt`

ls

- Description:
    - Lists directory contents of files and directories.
- Formula:
    - `ls + option + file or directory`

- Examples
  - It sorts the file by modification time
    - `ls -t`
  - Display One File Per Line
    - `ls -1`
  - To show long listing information about the file/directory.
    - `ls -l`

man

- Description:
  - Documentation files that describe Linux shell commands and options.
- Formula:
  - `man + command`
- Examples:
  - System calls, which are system request that programs make to the kernel.
    - `man kill`
  - Opens a specific man page for yast command
    - `man 5 yast`
  - Show the man page section of the passwd command.
    - `man -f passwd`

mkdir

- Description:
  - Used for creating a single directory or multiple directories.
- Formula:
  - `mkdir + the name of the directory`
- Examples
  - Create a directory in the present working directory
    - `mkdir wallpapers`
  - Create a directory with a parent directory at the same time.
    - `mkdir -p wallpapers/animated`

mv

- Description:
  - Moves and renames files or directory
- Formula:
  - `mv + source + destination` (moves and renames directories)
  - `mv + file/directory to rename + new name` (renames the file or directory without moving)
- Examples:
  - Moves a file from a directory to another.
    - `mv Downloads/homework.pdf Documents/`
  - Rename a file and move in the same command.
    - `Downloads/cis106homework.docx Document/new_cis106homework.docx`

tac

- Description:
  - Used to concatenate and print files in reverse.
- Formula:
  - `tac + option + file`
- Examples
  - It will print files in reverse.
    - `tac numbers1-10.txt`
  - This option attach the separator before instead of after.
    - `tac -b numbers1-10.txt numbers 11-21.txt`

tail

- Description:
  - Print the last N number of data of the given input.
- Formula:
  - `tail + option + file`
- Examples
  - How to see the last line in a file
    - `tail -1 /etc/passwd`
  - Without any option it display only the last 10 lines of the file specified.
    - `tail /etc/passwd`
  - Prints the last ‘num’ lines instead of last 10 lines.
    - `tail -n 3 state.txt`

touch

- Description:
  - Used to create files
- Formula:
  - `touch + file name`
- Examples
  - Create a file called list
    - `touch list`
  - Create a file with a space in the name.
    - `touch "list of games.txt`
  - To create several files.
    - `touch script.py games.doc performanc.pdf`

tr

- Description:
  - Translates or deletes characters.
- Formula:
  - `tr + option + set`
- Examples
  - To convert lower case characters to upper case.

        - `tr [:lower:] [:upper:] <greekfile`
    - How to squeeze a sequence of repetitive characters using -s option.
        - `tr -s " "`

tree

- Description:
    - Recursive directory listing program that produces a depth-indented listing of files.
- Formula:
    - `tree + option + file or directory`
- Examples
    - List files with entered pattern.
        - `tree -P sample.txt`
    - Display the tree hierarchy of a directory.
        - `tree -a sample.txt`
    - List those directories which have greater ‘N’ number of files/directories.
        - `tree --filelimit 3 sample.txt`

vim/nano

- Description:
    - Simple text editor, which improves the features and user-friendliness.
- Formula:
    - `nano + file`
- Examples
    - To create and open a new file.
        - `To create and open a new file`
    - To save a file.
        - `press Ctrl+o`

Question 2

- How to work with multiple terminals open?
    - Open one terminal the open another terminal and set them side by side. Or user tillix and split the terminal as needed.
- How to work with manual pages?
    - Type man on the command line followed by a space and the linux command, then a "man page" that describes the command appears. Using options you can set it up so that you see a specific page only from the man page.
- How to parse (search) for specific words in the manual page.
    - To search a specific man page section, use the -s option with the man command and the -k or -K option.
- How to redirect output (> and |).
    - The ‘>‘ symbol is used for output redirection.
- How to append the output of a command to a file.

    - When the notation > > filename is added to the end of a command, the output of the command is appended to the specified file name.
- How to use wildcards (For copying and moving multiple files at the same time).
    - To copy multiple files you can use wildcards (cp *. extension) having same pattern.
- How to use brace expansion (For creating entire directory structures in a single command)
    - To correctly-form a brace expansion to create entire directory structures from a single command you must contain unquoted opening and closing braces "{}" for said structure but make sure to add the mkdir command at the very start.