Questions for lab 1

- **What is a “shell program”?**

A shell program is a program that is written in the shell programming language, it provides a powerful way to automate tasks and perform system administration tasks in a Unix-like operating system, some examples include Bash (Bourne-Again SHell), Zsh (Z Shell), and more.

- **What can a shell program be used for(give examples)?**

Some of the uses we have from shell programs include these three things task automation, file manipulation, and program execution. Other examples include Web development, where shell script can be used for compiling code, running tests, and deploying applications.

- **How do you make a *shell(.sh) file* executable (what is the procedure)?**

Using linux we can Open
Applications > Accessories > Terminal.
Then find the .sh file. Use the ls and cd commands. ls will list the files and folders in the current folder. Then Run the .sh file. Once you can see for example script1.sh with ls run this:
./script.sh.

- **What is the difference between Interpreted code VS. Compiled code(in your own words)?**

interpreted code is known to be executed line by line, where it reads then executes the code, it is also more portable and might be slower, whereas compiled code is translated all at once into machine code before execution, where it's known to be less portable but faster.

- **What is the difference between running a program with and without *exec?***

The difference is that when Running a program without exec will run the program on the active/current running process, however, running a program with exec will cause the current process to be removed from the process table and will add a new entry of a new process in the table.

- ## What does the ps command do, and why is it useful?

ps commands can be used to view the information of all the currently running processes it also displays the CPU and memory usage, etc.
Ps commands are useful because it allows us to monitor the system's performance, resource usage and both manage and control processes on the system, we can also use this to determine which processes are running and which processes have less or greater priority.

- ## Why do we need 3 sets of file/folder permission for a file/folder?

We need 3 sets of file/folder permissions because Each set of permissions defines what actions can be performed on the file or folder by different categories of users,
These 3 sets are Owner permissions: Group permissions, and World or others permissions, sometimes we would like to give permission to 1 of these 3 sets, for example, the owner set only gives permission for the user to read, write, and execute the file or folder, therefore we need 3 sets of file/folder permission.