Part two:

1. Files

1. **Suppose the command `ls` were unavailable. What command could you use to view files in the current directory instead? Explain why the command works?**

   `dir` , this command works because it is meant to list the contents of a directory

2.
   - **What option do I need to supply to the `rm` command to ensure I am prompted before a file or directory is removed?**

     `-i`

   - **What option do I need to supply to the `rm` command to ensure I am not prompted before a file or directory is removed?**

     `--interactive=never`

   - **What happens if both options are used simultaneously - which one has precedence?**

     Always prompt (`-i`) has precedence

3. **What are some differences between the `rename` command and the `mv` command?**

   mv is used to move one or more files to a given location. It can rename a file, if there's only one file to move. While rename is use to rename multiple files at once.

4. **The `file` command, when run against an executable program shows `ELF 64-bit LSB executable`. What does `ELF` mean in this context?**

   `ELF` stands for Executable and Linkable Format. It is a standard that provides developers with a set of binary interface definitions that can be applied across multiple operating environments.

5. **Consider the empty file `test.txt`. Why does `ls -l test.txt` say the filesize is `0` but `du test.txt` says the file size is `1`?**

   This is because `ls -l` shows the size of the file, while `du` shows the amount of space the file uses up on disk, which is usually larger than the actual size of the file because it accounts for cluster space and any additional sectors where the file is stored.

2. Directories

1. **Find a better way to do the following:**

```
mkdir grandparent
mkdir grandparent/parent
mkdir grandparent/parent/child
```

   `mkdir -p grandparent/parent/child`

2.
    - **How do you create a directory with a space in it?**
      By surrounding the name in quotes eg: `mkdir 'with space'`
    - **How do you remove a directory with a space in it?**
      By surrounding the name in quotes eg: `rmdir 'with space'`

3. **How do I create a directory and set its permissions to `dr-x------` using only the `mkdir` command?**

   `mkdir -m u-w,u+rx,g-rwx,o-rwx directory`

4. **Suppose I was working in `/dir1/dir2/dir3` and I `cd` to `/dir4/dir5/dir6`. What is an easy way to get back to `/dir1/dir2/dir3` without using the explicit command `cd /dir1/dir2/dir3` (perhaps I forgot what the full path name was)?**

   `cd -` or `cd ../../..`

5. **Suppose I used `pushd` to add 10 folders to my stack. How do I quickly determine which folder is the 6th folder in my stack and how do I quickly cd to it?**

   You would use the command `pushd -5`. This would count the folders you navigated to starting from the first to last using an index that starts from 0. Then it would change directory to the 6th folder, putting this folder on top of the stack.

3. Permissions

1. **What option(s) do I need to add to `chmod` to print out what modifications are being made?**

   `-c or --changes`

2. **What option(s) do I need to add to `chown` to get `file2` ownership to simply match `file1`?**

   `chown  --reference=file1 file2`

3. **What option(s) do I need to add to the `stat` command to display file permission, user ownership, and group ownership only?**

   `stat file -c=%A%G%U` or `stat file --format=%A%G%U` or `stat file --printf=%A%G%U`

4. **How do you grant a group permission to write data to a file, but prevent the users in that group from moving, removing, or renaming the file?**

   You would first create the file with the group having write permissions. Then you would go up to the directory in which the file is contained and remove write permissions from the directory. So no files within the directory shall be deletable or movable by the group but the file itself shall be editable.

5. **When using `chmod` in symbolic mode, what happens if u/g/o/a are omitted. Eg. `chmod +x <filename>`**

   Changes are applied to all; the command behaves the same as `chmod a+x <filename>`