Question 3: Understanding file permissions

3.a Make sure you have a file called file.txt in your home directory (see 1.e). Now change directory to your home directory by typing “cd” at the command prompt.
Now type “`ls -l file.txt`”.
What are the file permissions set for the new file you just created?
Explain each field in the file permission set.

3.b create a directory testdir in your home directory using “`mkdir testdir`”. Explain the file permissions you see with `ls -ld testdir`

Why did you have to use the additional option “d”. Use man ls to find out. Also try without the -d option, i.e., `ls -l testdir`

If the output above is confusing, try the following:
create a file in testdir, by typing “`touch testdir/file`” Now, try `ls -l testdir` again.

3.c. `Open file.txt` and type in the shell script below (note that the spaces around [and] are important!). Then, save and quit your editor. Now, at the command prompt, type in “`sh file.txt <filename>`” (use some file name which exists, and then use a filename which does not exist)”. Explain what you see.

```
#!/bin/sh
if [ -f $1 ]
then
      cat $1
else
      echo "Sorry, cant find file"
fi
```

3.d Observe that you have just created a text file and executed it by passing it as an input to the shell (“sh”)! Now try to execute the file by itself, by typing
`./file.txt <filename>`
Did that work? Why or why not? Explain by looking at the file permissions for file.txt using the output of “ls -l file.txt” to see the permissions.

Now add execute permissions with `chmod u+x file.txt`.
Test that you can now execute file.txt with
`./file.txt <filename>`

3.e Move file.txt to testdir using “`mv file.txt testdir`”. (Make sure you are in your home directory before doing this!). Now list the testdir (`ls testdir`).
You can still execute file.txt. Except you have type in (`./testdir/file.txt <filename>`).
Check this works. The execute permissions are associated with the file rather than the directory.

3.f Recall that file permissions work differently for directories.
3.f.1 Remove ls permissions from `testdir` by typing the following from the command prompt (make sure you are in home directory)
`chmod u-r testdir`

3.f.2 Now try list the testdir (`ls testdir`) and report the results.

3.f.3 Of course, the ‘r’ bit is different from the ‘x’ bit, or the ‘w’.
For instance, if ‘x’ bit is set (it should be), you should still be able to traverse into testdir, and read its files or even execute them. Ensure this is true, by typing in:
(`./testdir/file.txt <filename>`)

You have just executed a file from a directory that you cannot read. So effectively, you cannot tell if the file file.txt is in testdir, but if it is there, you can execute it!

3.f.4 You can even remove files in testdir (because you have ‘w’ permission).
`rm testdir/file.txt`

3.f.5. You can even remove testdir itself, Ensure this with:
`rmdir testdir`
(Note you can only do this if testdir is empty. If it is not empty you can use “`rm -rf testdir`” to remove “testdir” and all its contents including other sub-directories. `rm -rf` is a dangerous command, be careful where you use it, so you don't lose files!)

3.f.6 ***Optional*** How does Linux ensure you have the permissions to remove testdir? Where is this permission stored? Explain your answer with a simple experiment. (hint: each directory can be considered as equivalent to a simple file, which maintains the mapping filename—> inode for each file in the dir.)