1. Create a folder called /tmp/myteam

```
[root@server ~]# mkdir /tmp/myteam
[root@server ~]#
```

2. Change its permissions to read only for the owner.

```
[root@server ~]# ls -l /tmp
total 0
srwxrwxrwx. 1 gdm  gdm   0 Nov 17 14:03 dbus-RkvCELTCpr
drwxr-xr-x. 2 root root  6 Nov 17 14:06 myteam
```

```
[root@server ~]# chmod 400 /tmp/myteam/
[root@server ~]# ls -l /tmp
total 0
srwxrwxrwx. 1 gdm  gdm   0 Nov 17 14:03 dbus-RkvCELTCpr
dr--------. 2 root root  6 Nov 17 14:06 myteam
```

3. Log out and log in by another user .Try to access the folder

```
[root@server ~]# su - ali
[ali@server ~]$ ls -l /tmp/myteam/
ls: cannot open directory '/tmp/myteam/': Permission denied
[ali@server ~]$
```

4. Using the command Line Change the permissions of /tmp/myteam/mycv dir to give owner read and write permissions and for group write and execute and execute only for the others (using chmod in 2 different ways )

```
[root@server ~]# mkdi
mkdict  mkdir
[root@server ~]# mkdir /tmp/myteam/mycv
[root@server ~]# ls -l /tmp/myteam/
total 0
drwxr-xr-x. 2 root root 6 Nov 17 14:17 mycv
[root@server ~]#
```

```
[root@server ~]# chmod u=rw,g=wx,o=x /tmp/myteam/mycv/
[root@server ~]#
[root@server ~]# ls -l /tmp/myteam/
total 0
drw--wx--x. 2 root root 6 Nov 17 14:17 mycv
[root@server ~]# chmod 631 /tmp/myteam/
[root@server ~]# chmod 631 /tmp/myteam/mycv/
[root@server ~]# ls -l /tmp/myteam/
total 0
drw--wx--x. 2 root root 6 Nov 17 14:17 mycv
[root@server ~]#
```

5. Change your default permissions to be as above (question 4).

```
[root@server ~]# umask 146
[root@server ~]#
```

6. What is the maximum permission a file can have, by default when it is just created? And what is that for directory.

Files: 666 (read/write for owner, group, and others).

Directories: 777 (read/write/execute for owner, group, and others).

7. Change your default permissions to be no permission to everyone then create a directory and a file to verify.

```
[root@server ~]# umask 777
[root@server ~]#
```

```
[root@server ~]# touch no_permission_file
[root@server ~]# mkdir no_permission_dir
```

```
[root@server ~]# ls -ld no_permission_file
----------. 1 root root 0 Nov 17 14:40 no_permission_file
[root@server ~]# ls -ld no_permission_dir
d---------. 2 root root 6 Nov 17 14:40 no_permission_dir
[root@server ~]#
```

8. Copy /etc/passwd file to your home directory. Note the permissions allowed to you before and after. Specify why?

```
[root@server ~]# cp /etc/passwd ~/
[root@server ~]# ls -l /etc/pa
pam.d/     papersize  passwd     passwd-
[root@server ~]# ls -l /etc/passwd ~/passwd
-rw-r--r--. 1 root root 2153 Nov 17 02:08 /etc/passwd
----------. 1 root root 2153 Nov 17 14:43 /root/passwd
[root@server ~]#
```

Before copy: /etc/passwd typically has 644 permissions (readable by everyone, writable only by the owner).

After copy: The copied file inherits the umask of the user.

9. What are the minimum permission needed for :
    1. Copy a directory (source and target)

       Source : r-x

       Target : wx

    2. Copy a file (source and target)

       Source : r

       Target : w

    3. Delete a file

       w
    4. Change to a directory

       x
    5. List a directory content

       r-x
    6. View a file content

       r
    7. Modify a file content

       r-w
10. Create a file with permission 444. Try to edit in it and to remove it? Note what happened.(notice write protection in Linux)

```
[root@server ~]# touch readfile
[root@server ~]# ls -l readfile
----------. 1 root root 0 Nov 17 15:06 readfile
[root@server ~]# chm
chmem  chmod
[root@server ~]# chmod 444 readfile
[root@server ~]# ls -l readfile
-r--r--r--. 1 root root 0 Nov 17 15:06 readfile
```

Edit: Denied because the file is read-only.

```
[ali@server ~]$ ls -l reaablefile
-r--r--r--. 1 root root 6 Nov 17 15:38 reaablefile
[ali@server ~]$ cat <<A>> reaablefile
> qwert
> A
-bash: reaablefile: Permission denied
[ali@server ~]$
```

**Delete**: Allowed if the parent directory has write permission

```
[ali@server ~]$ rm reaablefile
rm: remove write-protected regular file 'reaablefile'? y
[ali@server ~]$ ls
[ali@server ~]$
```

11. What is the difference between the “x” permission for a file and for a directory

    File: Execute permission allows the file to be run as a program/script.
    Directory: Execute permission allows entering the directory and accessing its contents if the file names are known.

12. What is the difference between the set-uid and set-gid?

    Set-UID: Executes a file with the owner's privileges.

    Set-GID:

    For files: Executes a file with the group's privileges.

    For directories: Files and directories created inside inherit the parent directory’s group.

13. Create a directory with sticky-bit and write permissions on, grant all the users to access the directory, will any user be able to create and delete files from the directory?

```
[root@server ~]# mkdir /tmp/sticky_dir
[root@server ~]# ls -ld /tmp/sticky_dir
drwxr-xr-x. 2 root root 6 Nov 17 15:46 /tmp/sticky_dir
[root@server ~]# chm
chmem  chmod
[root@server ~]# chmod 1777 /tmp/sticky_dir
[root@server ~]# ls -ld /tmp/sticky_dir
drwxrwxrwt. 2 root root 6 Nov 17 15:46 /tmp/sticky_dir
[root@server ~]#
```

14. Create a directory with set-gid permission, what do you notice when you create a new file or a directory

```
[root@server ~]# mkdir /tmp/sgid_dir
[root@server ~]# ls -ld /tmp/sgid_dir
drwxr-xr-x. 2 root root 6 Nov 17 15:48 /tmp/sgid_dir
[root@server ~]# chmod 2777 /tmp/sgid_dir/
[root@server ~]# ls -ld /tmp/sgid_dir
drwxrwsrwx. 2 root root 6 Nov 17 15:48 /tmp/sgid_dir
[root@server ~]#
```

Effect: Files and directories created inside inherit the group ownership of the parent directory rather than the creator's primary group.