Assignment 1- Basic Commands

echo commands

1) shreya@shreya-VirtualBox:~$ echo helloworld

   helloworld

2) shreya@shreya-VirtualBox:~$ echo *

   Desktop Documents Downloads Music Pictures Public Templates Videos

3) shreya@shreya-VirtualBox:~$ echo

   (blank line)

---

date commands

1) shreya@shreya-VirtualBox:~$ date +'%d-%m-%Y'

   22-08-2020

2) shreya@shreya-VirtualBox:~$ date +'%T'

   19:36:11

3) shreya@shreya-VirtualBox:~$ date +'%A, %d/%m/%Y'

   Saturday, 22/08/2020

cal commands

1) shreya@shreya-VirtualBox:~$ cal -1

```
      August 2020
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

2) shreya@shreya-VirtualBox:~$ cal 2020

```
                                2020

      January               February              March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4                  1      1  2  3  4  5  6  7
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   8  9 10 11 12 13 14
12 13 14 15 16 17 18   9 10 11 12 13 14 15  15 16 17 18 19 20 21
19 20 21 22 23 24 25  16 17 18 19 20 21 22  22 23 24 25 26 27 28
26 27 28 29 30 31     23 24 25 26 27 28 29  29 30 31
```

April

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

May

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

June

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

July

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

August

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

September

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

```
      October                November                December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr
Sa
                 1  2  3   1  2  3  4  5  6  7          1  2  3  4  5
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   6  7  8  9 10 11
12
11 12 13 14 15 16 17  15 16 17 18 19 20 21  13 14 15 16 17 18
19
18 19 20 21 22 23 24  22 23 24 25 26 27 28  20 21 22 23 24 25
26
25 26 27 28 29 30 31  29 30                 27 28 29 30 31
```

3) shreya@shreya-VirtualBox:~$ cal 8 2020

```
      August 2020
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
                30 31
```

---

who commands

1) shreya@shreya-VirtualBox:~$ who

```
shreya    :0           2020-08-22 19:31 (:0)
```

2) shreya@shreya-VirtualBox:~$ whoami

```
shreya
```

---

uname commands

1) shreya@shreya-VirtualBox:~$ uname -a

```
Linux shreya-VirtualBox 5.4.0-42-generic #46-Ubuntu SMP Fri Jul 10
00:24:02 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

2) shreya@shreya-VirtualBox:~$ uname –s

```
Linux
```

3) shreya@shreya-VirtualBox:~$ uname -r

```
5.4.0-42-generic
```

4) shreya@shreya-VirtualBox:~$ uname -sr

```
Linux 5.4.0-42-generic
```

---

shreya@shreya-VirtualBox:~$ df -h

```
Filesystem      Size  Used Avail Use% Mounted on
udev            2.5G     0  2.5G   0% /dev
tmpfs           500M  1.4M  498M   1% /run
```

```
/dev/sda6        15G  5.7G  7.9G  42% /
tmpfs           2.5G     0  2.5G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           2.5G     0  2.5G   0% /sys/fs/cgroup
/dev/sda5       511M  4.0K  511M   1% /boot/efi
tmpfs           500M  104K  500M   1% /run/user/1000
/dev/loop0       30M   30M     0 100% /snap/snapd/8542
/dev/loop1       55M   55M     0 100% /snap/core18/1880
/dev/loop2      256M  256M     0 100% /snap/gnome-3-34-1804/36
/dev/loop3       63M   63M     0 100% /snap/gtk-common-themes/1506
/dev/loop4       50M   50M     0 100% /snap/snap-store/467
```

---

history

```
shreya@shreya-VirtualBox:~$ history
    1  echo helloworld
    2  echo *
    3  echo
    4  date
    5  date +'%d-%m-%Y'
    6  man date
    7  date +'%T %d/%m/%Y'
    8  date +'%T'
    9  man date
```

```
   10  date +'%A, %d/%m/%Y'
   11  cal -1
   12  cal 2020
   13  cal 8 2020
   14  who
   15  whoami
   16  uname -a
   17  uname -s
   18  uname -r
   19  uname -sr
   20  df -h
   21  history
```

---

cat commands

1) Creating a file

```
shreya@shreya-VirtualBox:~$ cat>lockdown_story.txt
hi
this is lockdown
started in march 2020
```

2) Viewing the contents of the file

```
shreya@shreya-VirtualBox:~$ cat lockdown_story.txt
hi
```

this is lockdown

started in march 2020

---

ls commands

1) Listing contents of home directory with long format

```
shreya@shreya-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Desktop
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Documents
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Downloads
-rw-rw-r-- 1 shreya shreya   42 Aug 22 19:49 lockdown_story.txt
-rw-rw-r-- 1 shreya shreya   33 Aug 22 19:47 lockdownstory.txt
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Music
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Pictures
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Public
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Templates
drwxr-xr-x 2 shreya shreya 4096 Aug 22 19:31 Videos
```

2) Listing hidden files in home directory

```
shreya@shreya-VirtualBox:~$ ls -a
.             .config    .local              Pictures   Videos
..            Desktop    lockdown_story.txt  .profile
```

```
.bash_logout  Documents  lockdownstory.txt   Public
.bashrc       Downloads  .mozilla            .ssh
.cache        .gnupg     Music               Templates
```

3) Listing home directory files in recursive list

```
shreya@shreya-VirtualBox:~$ ls -R
.:
Desktop    Downloads           lockdownstory.txt  Pictures
Templates
Documents  lockdown_story.txt  Music              Public    Videos

./Desktop:
./Documents:
./Downloads:
./Music:
./Pictures:
./Public:
./Templates:
./Videos:
```

4) Listing home directory files : modified time order

```
shreya@shreya-VirtualBox:~$ ls -t
lockdown_story.txt  Desktop    Downloads  Pictures  Templates
lockdownstory.txt   Documents  Music      Public    Videos
```

---

1) Move to root directory

```
shreya@shreya-VirtualBox:~$ cd /
shreya@shreya-VirtualBox:/$
```

2) Files in root directory

```
shreya@shreya-VirtualBox:/$ ls
bin    dev   lib    libx32      mnt   root  snap      sys  var
boot   etc   lib32  lost+found  opt   run   srv       tmp
cdrom  home  lib64  media       proc  sbin  swapfile  usr
```

3) Move to home directory

```
shreya@shreya-VirtualBox:/$ cd ~
shreya@shreya-VirtualBox:~$
```

4) Display home directory

```
shreya@shreya-VirtualBox:~$ ls
Desktop    Downloads           lockdownstory.txt  Pictures
Templates
Documents  lockdown_story.txt  Music              Public    Videos
```

5) Create directory myfolder

```
shreya@shreya-VirtualBox:~$ mkdir myfolder
```

6) Move to myfolder

```
shreya@shreya-VirtualBox:~$ cd myfolder
```

7) Creating directory structure and viewing the structure using tree command

```
shreya@shreya-VirtualBox:~/myfolder$ mkdir Domain

shreya@shreya-VirtualBox:~/myfolder$ cd Domain

shreya@shreya-VirtualBox:~/myfolder/Domain$ mkdir IOT Dataanalytics Computervision

shreya@shreya-VirtualBox:~/myfolder/Domain$ cd IOT

shreya@shreya-VirtualBox:~/myfolder/Domain/IOT$ mkdir Cloudcomputing

shreya@shreya-VirtualBox:~/myfolder/Domain/IOT$ cd ..

shreya@shreya-VirtualBox:~/myfolder/Domain$ cd Dataanalytics

shreya@shreya-VirtualBox:~/myfolder/Domain/Dataanalytics$ mkdir Artificialintelligence
Datavisualization Cloudcomputing

shreya@shreya-VirtualBox:~/myfolder/Domain/Dataanalytics$ cd ..

shreya@shreya-VirtualBox:~/myfolder/Domain$ cd Computervision

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision$ mkdir Imageprocessing
Machinelearning Deeplearning

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision$ cd Imageprocessing

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Imageprocessing$ mkdir
Objectdetection Objectrecognition

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Imageprocessing$ cd
Objectdetection

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Imageprocessing/Objectdetection$
cat>Objdet1.txt

hi objectdet1

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Imageprocessing/Objectdetection$
cat>Objdet2.txt

hi objectdet2
```

```
shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Imageprocessing/Objectdetection$
cd ..

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Imageprocessing$ cd ..

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision$ cd Machinelearning

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Machinelearning$ mkdir
Classification Prediction

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Machinelearning$ cd
Classification

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Machinelearning/Classification$
cat>SVM.txt

hii SVM

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Machinelearning/Classification$
cat>Naivebayes.txt

hiii naivebayes

shreya@shreya-VirtualBox:~/myfolder/Domain/Computervision/Machinelearning/Classification$
cd ~
```

```
shreya@shreya-VirtualBox:~$ tree

├── Desktop

├── Documents

├── Downloads

├── lockdown_story.txt

├── lockdownstory.txt

├── Music
```

```
├── myfolder
│   └── Domain
│       ├── Computervision
│       │   ├── Deeplearning
│       │   ├── Imageprocessing
│       │   │   ├── Objectdetection
│       │   │   │   ├── Objdet1.txt
│       │   │   │   └── Objdet2.txt
│       │   │   └── Objectrecognition
│       │   └── Machinelearning
│       │       ├── Classification
│       │       │   ├── Naivebayes.txt
│       │       │   └── SVM.txt
│       │       └── Prediction
│       ├── Dataanalytics
│       │   ├── Artificialintelligence
│       │   ├── Cloudcomputing
│       │   └── Datavisualization
│       └── IOT
│           └── Cloudcomputing
├── Pictures
├── Public
├── Templates
└── Videos

24 directories, 6 files
```

Relative and absolute path commands

1) Moving to a directory using absolute path

```
shreya@shreya-VirtualBox:~/myfolder/Domain/Dataanalytics$ cd /

shreya@shreya-VirtualBox:/$ cd
~/myfolder/Domain/Computervision/Machinelearning/Classification

shreya@shreya-
VirtualBox:~/myfolder/Domain/Computervision/Machinelearning/Classificatio
n$
```

2) Moving to a directory using relative path

```
shreya@shreya-
VirtualBox:~/myfolder/Domain/Computervision/Machinelearning$ cd
~/myfolder/Domain/Computervision/Machinelearning/Prediction/

shreya@shreya-
VirtualBox:~/myfolder/Domain/Computervision/Machinelearning/Prediction$
```

3) Moving to myfolder

```
shreya@shreya-
VirtualBox:~/myfolder/Domain/Computervision/Machinelearning/Prediction$
cd ~/myfolder

shreya@shreya-VirtualBox:~/myfolder$
```

---

rmdir commands

1) Creating cloud1.txt

```
shreya@shreya-VirtualBox:~/myfolder$ cd Domain/IOT/Cloudcomputing

shreya@shreya-VirtualBox:~/myfolder/Domain/IOT/Cloudcomputing$
cat>cloud1.txt
```

hii cloud computing

2) Removing Cloudcomputing

shreya@shreya-VirtualBox:~/myfolder/Domain/IOT$ rmdir Cloudcomputing

rmdir: failed to remove 'Cloudcomputing': Directory not empty

3) Removing Cloudcomputing in recursive mode

shreya@shreya-VirtualBox:~/myfolder/Domain/IOT$ rmdir –r Cloudcomputing

---

copy commands

1) Copy the directory Classification along with its files in the Dataanalytics directory

shreya@shreya-VirtualBox:~$ cp -R ~/myfolder/Domain/Computervision/Machinelearning/Classification ~/myfolder/Domain/Dataanalytics

2) List the contents of Svm.txt present in the Dataanalytics directory

shreya@shreya-VirtualBox:~$ cat myfolder/Domain/Dataanalytics/Classification/SVM.txt

hii SVM

3) shreya@shreya-VirtualBox:~/myfolder/Domain$ ls -R

.:

```
Computervision  Dataanalytics  IOT

./Computervision:

Deeplearning  Imageprocessing  Machinelearning

./Computervision/Deeplearning:

./Computervision/Imageprocessing:

Objectdetection  Objectrecognition

./Computervision/Imageprocessing/Objectdetection:

Objdet1.txt  Objdet2.txt

./Computervision/Imageprocessing/Objectrecognition:

./Computervision/Machinelearning:

Classification  Prediction

./Computervision/Machinelearning/Classification:

Naivebayes.txt  SVM.txt

./Computervision/Machinelearning/Prediction:

./Dataanalytics:

Artificialintelligence  Classification  Datavisualization

./Dataanalytics/Artificialintelligence:

./Dataanalytics/Classification:

Naivebayes.txt  SVM.txt

./Dataanalytics/Datavisualization:

./IOT:
```

---

mv commands

1) Renaming directories

```
shreya@shreya-VirtualBox:~$ mv ~/myfolder/Domain/Computervision
~/myfolder/Domain/CV
```

```
shreya@shreya-VirtualBox:~$ mv ~/myfolder/Domain/Dataanalytics
~/myfolder/Domain/DA
```

2) Moving files of one folder to another directory

```
shreya@shreya-VirtualBox:~$ mv
~/myfolder/Domain/CV/Imageprocessing/Objectdetection/*.txt
~/myfolder/Domain/CV/Imageprocessing/Objectrecognition
```

---

rm command

```
shreya@shreya-VirtualBox:~$ rm -r
~/myfolder/Domain/DA/Classification/*.txt
```

---

ln commands

1) Creating a hard link

```
shreya@shreya-VirtualBox:~$ ln
~/myfolder/Domain/CV/Machinelearning/Classification/SVM.txt
~/myfolder/Domain/CV/Machinelearning/Classification/Svm.txt
```

2) Listing Inodes of the files

```
shreya@shreya-
VirtualBox:~/myfolder/Domain/CV/Machinelearning/Classification$ ls -i
793358 Naivebayes.txt  793357 Svm.txt  793357 SVM.txt
```

3) Removing the hardlink

```
shreya@shreya-
VirtualBox:~/myfolder/Domain/CV/Machinelearning/Classification$ rm
Svm.txt
```

```
shreya@shreya-
VirtualBox:~/myfolder/Domain/CV/Machinelearning/Classification$ ls -i
793358 Naivebayes.txt  793357 SVM.txt
```

4) Creating hardlink for directory

```
shreya@shreya-VirtualBox:~$ ln ~/myfolder/Domain/DA ~/myfolder/Domain
ln: /home/shreya/myfolder/Domain/DA: hard link not allowed for directory
```

5) Creating a soft link for directory

```
shreya@shreya-VirtualBox:~$ ln -s ~/myfolder/Domain/DA ~/myfolder
```

6) Inode for original and shortcut

```
shreya@shreya-VirtualBox:~/myfolder/Domain$ ls -i
793343 CV  793342 DA  793340 IOT
shreya@shreya-VirtualBox:~/myfolder/Domain$ cd ..
```

```
shreya@shreya-VirtualBox:~/myfolder$ ls -i
795109 DA  786475 Domain
```

7) Removing directory

```
shreya@shreya-VirtualBox:~$ rm -r ~/myfolder/Domain/DA
```

8) Checking that shortcut has become a dangling link

```
shreya@shreya-VirtualBox:~$ cd ~/myfolder
shreya@shreya-VirtualBox:~/myfolder$ ls -i
795109 DA  786475 Domain
shreya@shreya-VirtualBox:~/myfolder$ cd DA
bash: cd: DA: No such file or directory
```

ASSIGNMENT - 2 File system

Exercise 1

1. Launch a terminal.

2. Create three directories named letters, reports and assignment under your home directory.

```
shreya@shreya-VirtualBox:~$ mkdir letters reports assignments

shreya@shreya-VirtualBox:~$ ls

assignments  Documents  letters  myfolder  Public   Templates

Desktop      Downloads  Music    Pictures  reports  Videos
```

3. Move to directory letters.

```
shreya@shreya-VirtualBox:~$ cd letters
```

4. Create two directories named friendly and formal under the letters directory.

```
shreya@shreya-VirtualBox:~/letters$ mkdir friendly formal

shreya@shreya-VirtualBox:~/letters$ ls

formal  friendly
```

5. Move to directory reports using only one command

```
shreya@shreya-VirtualBox:~/letters$ cd ../reports/
```

6. **Create three directories called personal, business, and school under the directory reports (use only one command).**

```
shreya@shreya-VirtualBox:~/reports$ mkdir personal business school
```

7. Create a directory called UNIX under the assignments directory. The directories in this step should be created without moving from the reports directory.

```
shreya@shreya-VirtualBox:~/reports$ mkdir ~/assignments/UNIX
```

8. Move to your home directory.

```
shreya@shreya-VirtualBox:~/assignments$ cd ~
```

9. Recursively list all of the directories you created

```
shreya@shreya-VirtualBox:~$ ls -R

.:

assignments  Documents  letters  myfolder  Public   Templates

Desktop      Downloads  Music    Pictures  reports  Videos

./assignments:

UNIX

./assignments/UNIX:

./letters:

formal  friendly

./letters/formal:

./letters/friendly:

./reports:

business  personal  school

./reports/business:

./reports/personal:

./reports/school:
```

10.Quit terminal

Exercise 2

1. Launch a terminal.

2. Recursively list the directories under your home directory

```
shreya@shreya-VirtualBox:~$ ls -R

.:

assignments  Documents  letters  myfolder  Public   Templates

Desktop      Downloads  Music    Pictures  reports  Videos

./assignments:

UNIX

./assignments/UNIX:

./letters:

formal  friendly

./letters/formal:

./letters/friendly:

./reports:

business  personal  school

./reports/business:

./reports/personal:

./reports/school:
```

3. Move to the UNIX directory.

```
shreya@shreya-VirtualBox:~$ cd ~/assignments/UNIX
```

4. Check your current directory.

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ pwd
```

/home/shreya/assignments/UNIX

5. Create a file named hw4 that contains short answers to at least five review questions in this chapter.

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ cat>hw4.txt

unix and linux are not the same

i am using ubuntu
```

6. Save the file (it should be saved under the UNIX directory).

7. Move to your home directory

```
shreya@shreya-VirtualBox:~$ cd ~
```

8. Print the content of hw4 from your home directory.

```
shreya@shreya-VirtualBox:~$ cat  assignments/UNIX/hw4.txt

unix and linux are not the same

i am using ubuntu
```

9. Make a copy of hw4 and call it hw4.bk.

```
shreya@shreya-VirtualBox:~$ cp ~/assignments/UNIX/hw4.txt
~/assignments/UNIX/hw4.bk
```

10. Store it under the same directory where hw4 is stored.

11. From your home directory, check to see if both files (hw4 and hw4.bk) exist.

```
shreya@shreya-VirtualBox:~$ ls -R

.:

assignments  Documents  letters  myfolder  Public   Templates

Desktop      Downloads  Music    Pictures  reports  Videos

./assignments:
```

UNIX

./assignments/UNIX:

hw4.bk hw4.txt

12. Move to the UNIX directory.

shreya@shreya-VirtualBox:~$ cd assignments/UNIX

13. Check your current working directory.

shreya@shreya-VirtualBox:~/assignments/UNIX$ pwd

/home/shreya/assignments/UNIX

14. Make a hard link to the hw4 file. The link should be under the UNIX subdirectory and be called hw4HL.

shreya@shreya-VirtualBox:~/assignments/UNIX$ ln hw4.txt hw4HL

shreya@shreya-VirtualBox:~/assignments/UNIX$ ls -li

total 12

795458 -rw-rw-r-- 1 shreya shreya 50 Aug 28 14:05 hw4.bk

796164 -rw-rw-r-- 2 shreya shreya 50 Aug 28 13:53 hw4HL

796164 -rw-rw-r-- 2 shreya shreya 50 Aug 28 13:53 hw4.txt

15. Make a soft link to hw4 called hw4SL and store it under the UNIX directory.

shreya@shreya-VirtualBox:~/assignments/UNIX$ ln -s hw4.txt hw4SL

16. Check the inode of hw4, hw4.bk, hw4HL, and hw4SL. Are all the same? Are all different? Explain how you determined the answer

shreya@shreya-VirtualBox:~/assignments/UNIX$ ls -li

total 12

795458 -rw-rw-r-- 1 shreya shreya 50 Aug 28 14:05 hw4.bk

796164 -rw-rw-r-- 2 shreya shreya 50 Aug 28 13:53 hw4HL

796167 lrwxrwxrwx 1 shreya shreya 7 Aug 28 14:10 hw4SL -> hw4.txt

796164 -rw-rw-r-- 2 shreya shreya 50 Aug 28 13:53 hw4.txt

The inode numbers of directories and files are used to differentiate between a hard link and a soft link. If inode numbers are same as the original file it's a hard link else it is a soft link

17. Use ls command to find the file types of hw4, hw4.bk, hw4HL and hw4SL. Explain your observation.

shreya@shreya-VirtualBox:~/assignments/UNIX$ ls --file-type

hw4.bk hw4HL hw4SL@ hw4.txt

Using ls command it is possible to determine the file types of all the files present in the present working directory.

In this case hw4.bk,hw4HL,hw4.txt is a copy,hardlink and the original file respectively. Soft link is followed by @.

18. Quit the terminal.

Exercise 3

1. Launch a terminal.

2. Create a backup directory in your home directory called backups.

shreya@shreya-VirtualBox:~$ cd backups

shreya@shreya-VirtualBox:~/backups$ ls

hw4.bk hw4HL hw4SL hw4.txt

3. Use the find command to find the pathnames of all of the files (hw4, hw4.bk, hw4HL, hw4SL) that you created in Exercise 2. All of them should be found using only one find command. The command must also copy all of them to the backups directory.

shreya@shreya-VirtualBox:~$ find ~/assignments/UNIX/ -type f -name 'hw4*' -exec cp {} ~/backups \;

4. Check the number of links and inode number of (hw4, hw4.bk, hw4HL, hw4SL). Make note of the results.

```
shreya@shreya-VirtualBox:~/backups$ ls -li

total 12

796170 -rw-rw-r-- 1 shreya shreya 50 Aug 28 14:38 hw4.bk

796175 -rw-rw-r-- 1 shreya shreya 50 Aug 28 14:38 hw4HL

796172 lrwxrwxrwx 1 shreya shreya  7 Aug 28 14:29 hw4SL -> hw4.txt

796178 -rw-rw-r-- 1 shreya shreya 50 Aug 28 14:38 hw4.txt
```

5. Delete the original hw4 file without moving from your home directory.

```
shreya@shreya-VirtualBox:~$ rm ~/assignments/UNIX/hw4.txt
```

6. Check the existence of hw4, hw4.bk, hw4HL, hw4SL.

```
shreya@shreya-VirtualBox:~$ cd ~/assignments/UNIX/

shreya@shreya-VirtualBox:~/assignments/UNIX$ ls -li

total 8

795458 -rw-rw-r-- 1 shreya shreya 50 Aug 28 14:05 hw4.bk

796164 -rw-rw-r-- 1 shreya shreya 50 Aug 28 13:53 hw4HL

796167 lrwxrwxrwx 1 shreya shreya  7 Aug 28 14:10 hw4SL -> hw4.txt
```

7. Check the contents of hw4, hw4.bk, hw4HL, hw4SL.

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ cat hw4.bk hw4.txt hw4HL.bk
hw4SL

unix and linux are not the same

i am using ubuntu

cat: hw4.txt: No such file or directory

unix and linux are not the same

i am using ubuntu
```

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ cd hw4SL

cat: hw4SL: No such file or director
```

8. Restore hw4 by making a copy of hw4.bk.

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ cp hw4.bk hw4

shreya@shreya-VirtualBox:~/assignments/UNIX$ ls

hw4  hw4.bk  hw4HL  hw4SL
```

9. You may have noticed that your soft link (hw4SL) contains garbage. Delete this file.

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ rm hw4SL

shreya@shreya-VirtualBox:~/assignments/UNIX$ ls

hw4  hw4.bk  hw4HL
```

10. Make a soft link to hw4 and save it as hw4SL under the same directory as it was.

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ ln -s hw4 hw4SL

shreya@shreya-VirtualBox:~/assignments/UNIX$ ls -li

total 12

793377 -rw-rw-r-- 1 shreya shreya 50 Sep  9 13:57 hw4

795458 -rw-rw-r-- 1 shreya shreya 50 Aug 28 14:05 hw4.bk

796164 -rw-rw-r-- 1 shreya shreya 50 Aug 28 13:53 hw4HL

795238 lrwxrwxrwx 1 shreya shreya  3 Sep  9 13:59 hw4SL -> hw4
```

11. List recursively all of your files and directories to confirm all operations.

```
shreya@shreya-VirtualBox:~$ ls -R

.:

./assignments:
```

```
UNIX


./assignments/UNIX:
hw4  hw4.bk  hw4HL  hw4SL
./backups:
hw4.bk  hw4HL  hw4SL  hw4.txt
```

12. Print the number files created given the date

```
shreya@shreya-VirtualBox:~$ ls -l |grep -c '^-'
7
shreya@shreya-VirtualBox:~$ ls -l |grep -c '^l'
0
shreya@shreya-VirtualBox:~$ ls -l |grep -c '^d'
19
```

13. Print the number of files created given the type

14. Quit the terminal.

Exercise 4

1. Launch a terminal.

2. Check which group or groups you belong to.

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ groups

shreya adm cdrom sudo dip plugdev lpadmin lxd sambashare
```

3. Use the umask command to set the default permission to 700. What is the default permission for files after this command?

```
shreya@shreya-VirtualBox:~/assignments/UNIX$ umask 077

shreya@shreya-VirtualBox:~/assignments/UNIX$ umask

0077
```

Ans. The default changes to 700 which means that the user has the permission to read,write,execute permissions whereas the users in the group and others cant access the files created

4. Create a directory named chapter4 under your home directory.

```
shreya@shreya-VirtualBox:~$ mkdir chapter4
```

5. Check the default permission of this directory. Is it 700?

```
shreya@shreya-VirtualBox:~$ cd chapter4

shreya@shreya-VirtualBox:~$ ls -l

total 108

drwxrwxr-x 3 shreya shreya 4096 Aug 28 13:40 assignments

-rw-rw-r-- 1 shreya shreya  322 Sep  3 09:32 a.txt

drwxrwxr-x 2 shreya shreya 4096 Aug 28 14:29 backups

drwx------ 2 shreya shreya 4096 Sep  9 22:48 chapter4
```

Ans. Yes. The default permission of the created directory is 700.

6. Create a directory under the chapter4 directory (without moving from your home directory) and name it session1.

```
shreya@shreya-VirtualBox:~$ cd chapter4/session1
```

7. Check the permission of this directory. Is it 700?

```
shreya@shreya-VirtualBox:~$ ls chapter4 -l

total 4

drwx------ 2 shreya shreya 4096 Sep  9 22:49 session1

shreya@shreya-VirtualBox:~/chapter4/session1$ touch hw41.txt

shreya@shreya-VirtualBox:~/chapter4/session1$ ls -l

total 0

-rw------- 1 shreya shreya 0 Sep  9 22:50 hw41.txt
```

Ans. Yes. The default permission of the created directory is 700.

8. Move to the session1 directory.

```
shreya@shreya-VirtualBox:~/$  cd ~/chapter4/session1
```

9. Create a file named hw41 under this directory. Save this file.

```
shreya@shreya-VirtualBox:~/chapter4/session1$ touch hw41.txt
```

10. Check the permission of this file. Is it 700 or 600? Why? Explain the difference between the permissions for files and directories.

```
shreya@shreya-VirtualBox:~/chapter4/session1$ ls -l
total 0
-rw------- 1 shreya shreya 0 Sep  2 15:00 hw41.txt
```

Ans. The permission of the file is 600 because a file has only read and write permissions. Files and Directories have difference in permission sets. Directories have three types of permissions whereas files have two types of permissions.

11. Do users in your group have any access to this file? Do other users (outside of your group) have any access to this file?

Ans. According to the changed user mask ,the members of the group as well as other people outside the group cant access the file because the permissions are not provided.

12. Change the permissions to allow users in your group only to copy this file to their own directories. Is there a need to change the permission of any directories? If yes, make necessary changes.

```
shreya@shreya-VirtualBox:~$ chmod 770 chapter4

shreya@shreya-VirtualBox:~$ ls -l

total 108

drwxrwxr-x 3 shreya shreya 4096 Aug 28 13:40 assignments

-rw-rw-r-- 1 shreya shreya  322 Sep  3 09:32 a.txt

drwxrwxr-x 2 shreya shreya 4096 Aug 28 14:29 backups

drwxrwx--- 3 shreya shreya 4096 Sep  9 22:49 chapter4

drwxrwxr-x 2 shreya shreya 4096 Sep  3 09:45 course

drwxrwxr-x 4 shreya shreya 4096 Aug 28 13:35 letters
```

13. Let a user in your group copy this file into her home directory. Was the copy success- ful? If not, find the reason and take the appropriate action(s) to correct it. Then try again with the other user in your group.

```
shreya@shreya-VirtualBox:~$ sudo adduser shrey1110

[sudo] password for shreya:

Adding user `shrey1110' ...

Adding new group `shrey1110' (1001) ...

Adding new user `shrey1110' (1001) with group `shrey1110' ...

Creating home directory `/home/shrey1110' ...
```

```
Copying files from `/etc/skel' ...

New password:

Retype new password:

passwd: password updated successfully

Changing the user information for shrey1110

Enter the new value, or press ENTER for the default

      Full Name []: shreyas

      Room Number []: 1

      Work Phone []: 23111

      Home Phone []: 9999

      Other []: 9999

Is the information correct? [Y/n] y

shreya@shreya-VirtualBox:~$ sudo cp

cp: missing file operand

Try 'cp --help' for more information.

shreya@shreya-VirtualBox:~$ sudo cp
/home/shreya/chapter4/session1/hw41.txt /home/shrey1110/chapter4

shreya@shreya-VirtualBox:~$
```

Ans. The file is copied in the home directory of the user suve0821 using the above method.

To create a new user, use the adduser command and repeat the same syntax to copy the file for a new user. As the permission of the file is 755(octal value for permission given above),the file is copied into the new user’s home directory

14. Quit the terminal.

Assignment 4: Filters

Exercise 1

1.Launch a terminal.
2. Use the cat command to create a file containing the following data. Name it Ch6S1F1.
Use tabs to separate the fields.
1425 Juan 14.25
4321 George 21.11
6781 Anna 16.77
1451 Ben 21.77
2277 Tuan 18.77

```
shreya@shreya-VirtualBox:~$ cat>Ch6S1F1

1425  Juan  14.25

4321  George 21.11

6781  Anna  16.77

1451  Ben   21.77

2277  Tuan  18.77
```

3. Use the cat command to display the file and check for accuracy.

```
shreya@shreya-VirtualBox:~$ cat -vet Ch6S1F1

1425^IJuan^I14.25$

4321^IGeorge^I21.11$

6781^IAnna^I16.77$

1451^IBen^I21.77$

2277^ITuan^I18.77$
```

4. Use the sort command to sort the file Ch6SlFl according to the first field. Call the sorted file Ch6S1F1 (same name).

```
shreya@shreya-VirtualBox:~$ sort Ch6S1F1 -o Ch6S1F1
```

5. Display the file Ch6S1F1.

```
shreya@shreya-VirtualBox:~$ cat Ch6S1F1

1425  Juan  14.25

1451  Ben   21.77

2277  Tuan  18.77

4321  George      21.11

6781  Anna  16.77
```

6. Use the cut and paste commands to swap fields 2 and 3 of Ch6S1F1. Call it Ch6S1F1.

```
shreya@shreya-VirtualBox:~$ cut -f 1,3 Ch6S1F1 | paste - >f.txt | cut -f
2 Ch6S1F1 |paste f.txt ->Ch6S1F1
```

7. Display the file Ch6S1F1.

```
shreya@shreya-VirtualBox:~$ cat Ch6S1F1

1425  14.25 Juan

4321  21.11 George

6781  16.77 Anna

1451  21.77 Ben

2277  18.77 Tuan
```

8. Quit the terminal.

```
shreya@shreya-VirtualBox:~$exit
```

Exercise 2

1 Launch a terminal.
2 Use the tail command to create and save the following file. Call it Ch6S2F1

PASSES ALL DATA FROM INPUT TO OUTPUT
PASSES ONLY SPECIFIED COLUMNS
PASSES NUMBER OF SPECIFIED LINES AT BEGINNING
COMBINES COLUMNS
ARRANGES DATA IN SEQUENCE
PASSES NUMBER OF SPECIFIED LINES AT THE END OF DATA
TRANSLATES ONE OR MORE CHARACTERS
DELETES DUPLICATE LINES
COUNTS CHARACTERS} WORDS, OR'LINES
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```
shreya@shreya-VirtualBox:~$ tail>Ch6S2F1
```

PASSES ALL DATA FROM INPUT TO OUTPUT

PASSES ONLY SPECIFIED COLUMNS

PASSES NUMBER OF SPECIFIED LINES AT BEGINNING

COMBINES COLUMNS

ARRANGES DATA IN SEQUENCE

PASSES NUMBER OF SPECIFIED LINES AT THE END OF DATA

TRANSLATES ONE OR MORE CHARACTERS

DELETES DUPLICATE LINES

COUNT CHARACTERS} WORDS, OR'LINES

ABCDEFGHIJKLMNOPQRSTUVWXYZ

3 Use the cat command to view its contents.

```
shreya@shreya-VirtualBox:~$ cat Ch6S2F1

PASSES ALL DATA FROM INPUT TO OUTPUT

PASSES ONLY SPECIFIED COLUMNS

PASSES NUMBER OF SPECIFIED LINES AT BEGINNING

COMBINES COLUMNS

ARRANGES DATA IN SEQUENCE

PASSES NUMBER OF SPECIFIED LINES AT THE END OF DATA

TRANSLATES ONE OR MORE CHARACTERS

DELETES DUPLICATE LINES

COUNT CHARACTERS} WORDS, OR'LINES

ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

4 Encrypt this file using the following steps:
(a) Reverse the file line by line (the last line becomes the first, the line before the last line becomes the second, and so on).
(b) Call the file Ch6S2F1Encr.

```
shreya@shreya-VirtualBox:~$ tac Ch6S2F1 |cat› Ch6S2F1Encr
```

5. Use the cat command to view its contents.

```
shreya@shreya-VirtualBox:~$ cat Ch6S2F1Encr

ABCDEFGHIJKLMNOPQRSTUVWXYZ

COUNT CHARACTERS} WORDS, OR'LINES

DELETES DUPLICATE LINES

TRANSLATES ONE OR MORE CHARACTERS

PASSES NUMBER OF SPECIFIED LINES AT THE END OF DATA
```

ARRANGES DATA IN SEQUENCE

COMBINES COLUMNS

PASSES NUMBER OF SPECIFIED LINES AT BEGINNING

PASSES ONLY SPECIFIED COLUMNS

PASSES ALL DATA FROM INPUT TO OUTPUT

7. Decrypt the file (reverse the encryption steps). Call it Ch6S2F1 (original name).

shreya@shreya-VirtualBox:~$ tac Ch6S2F1Encr | cat>Ch6S2F1

8.Display the file Ch6S1F1.

shreya@shreya-VirtualBox:~$ cat Ch6S2F1

PASSES ALL DATA FROM INPUT TO OUTPUT

PASSES ONLY SPECIFIED COLUMNS

PASSES NUMBER OF SPECIFIED LINES AT BEGINNING

COMBINES COLUMNS

ARRANGES DATA IN SEQUENCE

PASSES NUMBER OF SPECIFIED LINES AT THE END OF DATA

TRANSLATES ONE OR MORE CHARACTERS

DELETES DUPLICATE LINES

COUNT CHARACTERS} WORDS, OR'LINES

ABCDEFGHIJKLMNOPQRSTUVWXYZ

9. Quit the terminal.

shreya@shreya-VirtualBox:~$exit

Exercise 3

1.Launch a terminal.
2.Use the cat command to create and save the folloWing file. Call it Ch6S3F1.
ALPHABETICAL FACTS.
THE FIRST THREE LETTERS ARE ABC.
THE MEDIAN LETTERS ARE MN.
THE LAST THREE LETTERS ARE XYZ.
THE FIRST WORD IN MY DICTIONARY IS AAL.
THE LAST WORD IN MY DICTIONARY IS ZYTHUM.
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
THE LAST LETTER MAY BE PRONOUNCED ZEE OR ZED.
THE FIRST GREEK LETTER IS ALPHA.
THE LAST GREEK LETTER IS OMEGA.

```
shreya@shreya-VirtualBox:~$ cat>Ch6S3F1

ALPHABETICAL FACTS.
THE FIRST THREE LETTERS ARE ABC.
THE MEDIAN LETTERS ARE MN.
THE LAST THREE LETTERS ARE XYZ.
THE FIRST WORD IN MY DICTIONARY IS AAL.
THE LAST WORD IN MY DICTIONARY IS ZYTHUM.
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
THE LAST LETTER MAY BE PRONOUNCED ZEE OR ZED.
THE FIRST GREEK LETTER IS ALPHA.
THE LAST GREEK LETTER IS OMEGA.
```

3. Use the cat command to check the contents.

```
shreya@shreya-VirtualBox:~$ cat –vet Ch6S3F1

ALPHABETICAL FACTS.$
THE FIRST THREE LETTERS ARE ABC.$
THE MEDIAN LETTERS ARE MN.$
THE LAST THREE LETTERS ARE XYZ.$
THE FIRST WORD IN MY DICTIONARY IS AAL.$
THE LAST WORD IN MY DICTIONARY IS ZYTHUM.$
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.$
THE LAST LETTER MAY BE PRONOUNCED ZEE OR ZED.$
THE FIRST GREEK LETTER IS ALPHA.$
```

THE LAST GREEK LETTER IS OMEGA.$

4. Display the file.

ALPHABETICAL FACTS.
THE FIRST THREE LETTERS ARE ABC.
THE MEDIAN LETTERS ARE MN.
THE LAST THREE LETTERS ARE XYZ.
THE FIRST WORD IN MY DICTIONARY IS AAL.
THE LAST WORD IN MY DICTIONARY IS ZYTHUM.
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
THE LAST LETTER MAY BE PRONOUNCED ZEE OR ZED.
THE FIRST GREEK LETTER IS ALPHA.
THE LAST GREEK LETTER IS OMEGA.

5. Using the tr command, encrypt this file by shifting each letter five characters to the end of the character set.
For example, A becomes F, B becomes G, and soon. The end of the alphabet will wrap around. For example, Y becomes D and Z becomes E. Spaces and newlines would be preserved. This is called Caesarian encryption because it was invented by Julius Caesar. Call the encrypted file Ch6S3F1Encr.

shreya@shreya-VirtualBox:~$ cat Ch6S3F1|tr "[A-U]VWXYZ" "[F-Z]ABCDE" |cat>Ch6S3F1Encr

6. Use the cat command to check the contents of the encrypted file.

shreya@shreya-VirtualBox:~$ cat –vet Ch6S3F1Encr

FQUMFGJYNHFQ KFHYX.$

YMJ KNWXY YMWJJ QJYYJWX FWJ FGH.$

YMJ RJINFS QJYYJWX FWJ RS.$

YMJ QFXY YMWJJ QJYJYWX FWJ CDE.$

YMJ KNWXY BTWI NS RD INHYNTSFWD NX FFQ.$

YMJ QFXY BTWI NS RD INHYNTSFWD NX EDYMZR.$

YMJ VZNHP GWTBS KTC OZRUX TAJW YMJ QFED ITL.$

YMJ QFXY QJYJYW RFD GJ UWTSTZSHJI EJJ TW EJI.$

YMJ KNWXY LWJJP QJYYJW NX FQUMF.$

YMJ QFXY LWJJP QJYYJW NX TRJLF.$

7. Display the file.

shreya@shreya-VirtualBox:~$ cat Ch6S3F1Encr

FQUMFGJYNHFQ KFHYX.

YMJ KNWXY YMWJJ QJYYJWX FWJ FGH.

YMJ RJINFS QJYYJWX FWJ RS.

YMJ QFXY YMWJJ QJYJYWX FWJ CDE.

YMJ KNWXY BTWI NS RD INHYNTSFWD NX FFQ.

YMJ QFXY BTWI NS RD INHYNTSFWD NX EDYMZR.

YMJ VZNHP GWTBS KTC OZRUX TAJW YMJ QFED ITL.

YMJ QFXY QJYJYW RFD GJ UWTSTZSHJI EJJ TW EJI.

YMJ KNWXY LWJJP QJYYJW NX FQUMF.

YMJ QFXY LWJJP QJYYJW NX TRJLF.

8. Now use decryption (reverse strategy) to decrypt the file. Call the
new file Ch6S3F1 (original name).

shreya@shreya-VirtualBox:~$ cat Ch6S3F1Encr|tr "[F-Z]ABCDE" "[A-U]VWXYZ" |cat>Ch6S3F1

9. Use the cat command to look at the contents of the file. Is it the same as the original file?

shreya@shreya-VirtualBox:~$ cat –vet Ch6S3F1

ALPHABETICAL FACTS.$

THE FIRST THREE LETTERS ARE ABC.$

THE MEDIAN LETTERS ARE MN.$

THE LAST THREE LETETRS ARE XYZ.$

THE FIRST WORD IN MY DICTIONARY IS AAL.$

THE LAST WORD IN MY DICTIONARY IS ZYTHUM.$

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.$

THE LAST LETETR MAY BE PRONOUNCED ZEE OR ZED.$

THE FIRST GREEK LETTER IS ALPHA.$

THE LAST GREEK LETTER IS OMEGA.$

Yes it is the same as the original file.

10.Display the file.

shreya@shreya-VirtualBox:~$ cat Ch6S3F1

ALPHABETICAL FACTS.

THE FIRST THREE LETTERS ARE ABC.

THE MEDIAN LETTERS ARE MN.

THE LAST THREE LETETRS ARE XYZ.

THE FIRST WORD IN MY DICTIONARY IS AAL.

THE LAST WORD IN MY DICTIONARY IS ZYTHUM.

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.

THE LAST LETETR MAY BE PRONOUNCED ZEE OR ZED.

THE FIRST GREEK LETTER IS ALPHA.

THE LAST GREEK LETTER IS OMEGA.

11. Quit the terminal.

```
shreya@shreya-VirtualBox:~$exit
```

Exercise 4

1.Launch a terminal.

2.Use the cat command to create and save the followmg file. Do not type the headings. Call it Ch6S4F1.

```
shreya@shreya-VirtualBox:~$ cat>Ch6S4F1

1420  12.56  45

3456  14.56  22

2341  45.12  34

1122  23.55  28

1443  23.23  19

2351  67.90   56

8001  7.00   14
```

3.Use the cat command to check its contents.

```
shreya@shreya-VirtualBox:~$ cat -vet Ch6S4F1

1420^I12.56^I45$

3456^I14.56^I22$

2341^I45.12^I34$

1122^I23.55^I28$

1443^I23.23^I19$

2351^I67.90^I56$

8001^I7.00^I14$
```

4. Display the file.

```
shreya@shreya-VirtualBox:~$ cat Ch6S4F1

1420  12.56 45
```

```
3456  14.56 22

2341  45.12 34

1122  23.55 28

1443  23.23 19

2351  67.90 56

8001  7.00  14
```

5.Use a command to show the number of workers.

```
shreya@shreya-VirtualBox:~$ wc -l Ch6S4F1

7 Ch6S4F1
```

6.Use a command to sort the file based on id.

```
shreya@shreya-VirtualBox:~$ sort Ch6S4F1 -o Ch6S4F1

shreya@shreya-VirtualBox:~$ cat Ch6S4F1

1122  23.55   28

1420  12.56   45

1443  23.23   19

2341  45.12   34

2351  67.90  56

3456  14.56   22
```

7.Use one single command to show the worker who is paid the highest hourly rate.

```
shreya@shreya-VirtualBox:~$ sort -n +1 -2 Ch6S4F1 |tail -n1

2351  67.90   56
```

8.Use one single command to show the worker who worked more than anybody else. The command should show only the id of the worker.

```
shreya@shreya-VirtualBox:~$ sort -n +2 -3 Ch6S4F1 |tail -n1|cut -f
1

2351
```

9.Quit the terminal.

```
shreya@shreya-VirtualBox:~$exit
```

Exercise 5

1. Launch a terminal.
2. Use the cat command to copy file Ch6S4F1 and name it Ch6S5F1.

```
shreya@shreya-VirtualBox:~$ cat Ch6S4F1 >Ch6S5F1
```

3.Use the cat command to create and save the following file. Do not type the headings. Call it Ch6S5F2.

```
shreya@shreya-VirtualBox:~$ cat>Ch6S5F2

1420  12.56 45

2456  14.56 22

2341  45.12 34

1322  23.56 28

1443  23.23 19

2351  67.90 56

3467  56.90 14
```

4. Use the cat command to check the contents of both files.
5. Display both files.

```
shreya@shreya-VirtualBox:~$ cat Ch6S5F1
```

```
1122  23.55 28

1420  12.56 45

1443  23.23 19

2341  45.12 34

2351  67.90   56

3456  14.56 22

8001  7.00  14
```

```
shreya@shreya-VirtualBox:~$ cat Ch6S5F2

1420  12.56 45

2456  14.56 22

2341  45.12 34

1322  23.56 28

1443  23.23 19

2351  67.90 56

3467  56.90 14
```

6.Sort each file using the file id as the sort key. Save the sorted files as separate files.

```
shreya@shreya-VirtualBox:~$ sort -m Ch6S5F1 Ch6S5F2 -o Ch6S5F3
```

7.Use a command to merge two files created in step 7 on the id field. Call the new file Ch6S5F3.

```
shreya@shreya-VirtualBox:~$ cat Ch6S5F3

1122  23.55 28

1322  23.56 28
```

1420 12.56 45

1420 12.56 45

1443 23.23 19

1443 23.23 19

2341 45.12 34

2341 45.12 34

2351 67.90 56

2351 67.90 56

2456 14.56 22

3456 14.56 22

3467 56.90 14

8001 7.00 14

8. Use a command to remove the duplicate from the file and Save it without renaming it.

```
shreya@shreya-VirtualBox:~$ sort -u Ch6S5F3 -o Ch6S5F3
```

9. Display the file.

```
shreya@shreya-VirtualBox:~$ cat Ch6S5F3
```

1122 23.55 28

1322 23.56 28

1420 12.56 45

1443 23.23 19

2341 45.12 34

2341 45.12 34

2351 67.90 56

2351 67.90 56

2456 14.56 22

3456 14.56 22

3467 56.90 14

8001 7.00 14

10. Quit the terminal.
shreya@shreya-VirtualBox:~$exit

Exercise 6

1. Launch a terminal.
2. Use the cat command to create and save the following file. Do not type the headings. Call it C6S6F1.
Department Course Session Enrollment
CIS 15 1 45
CIS 54 1 20
BUS 34 2 20
ENG 11 2 89
CIS 45 1 38
MTH 35 1 56
MTH 35 2 41
PE 17 2 25
CIS 54 2 67

shreya@shreya-VirtualBox:~$ cat>C6F6F1

CIS 15 1 45

CIS 54 1 20

BUS 34 2 20

ENG 11 2 89

CIS 45 1 38

MTH 35 1 56

MTH 35 2 41

```
PE    17    2     25
CIS   54    2     67
```

3. Use the cat command to check the contents of the file.

```
shreya@shreya-VirtualBox:~$ cat -vet C6F6F1
CIS^I15^I1^I45$
CIS^I54^I1^I20$
BUS^I34^I2^I20$
ENG^I11^I2^I89$
CIS^I45^I1^I38$
MTH^I35^I1^I56$
MTH^I35^I2^I41$
PE^I17^I2^I25$
CIS^I54^I2^I67$
```

4. Display the file.

```
shreya@shreya-VirtualBox:~$ cat C6F6F1
CIS   15    1     45
CIS   54    1     20
BUS   34    2     20
ENG   11    2     89
CIS   45    1     38
MTH   35    1     56
MTH   35    2     41
```

```
PE    17    2     25

CIS   54    2     67
```

5.Use one command to sort the file on department course and session. The resulting file should be ordered first by department; within equal departments, it should be ordered on course; and within equal courses, it should be ordered by session. Hint: use three field specifiers:
department, course, and session.

```
shreya@shreya-VirtualBox:~$ sort -b +0 -1 +1n -2 +2n -3 C6F6F1 -o
C6F6F1
```

6.Display the file.

```
shreya@shreya-VirtualBox:~$ cat C6F6F1

BUS   34    2     20

CIS   15    1     45

CIS   45    1     38

CIS   54    1     20

CIS   54    2     67

ENG   11    2     89

MTH   35    1     56

MTH   35    2     41

PE    17    2     25
```

7.Quit the terminal.
```
shreya@shreya-VirtualBox:~$exit
```

Exercise 7

4.Launch a terminal.
5.Make a copy of /etc /passwd file and save it in a file called Ch6S7F1.

```
shreya@shreya-VirtualBox:~$ cp /etc/passwd Ch6S7F1
```

3.Use a command to count the number of users in this file. Make a note of it.

```
shreya@shreya-VirtualBox:~$ wc -l Ch6S7F1

47 Ch6S7F1
```

4.Cut the file so that each line has only two columns: login name (column 1) and user id (column 3). Call the new file Ch6S7F2.

```
shreya@shreya-VirtualBox:~$ cut -d ':' -f1 Ch6S7F1 |cat>ch6f7s1

shreya@shreya-VirtualBox:~$ cut -d ':' -f3 Ch6S7F1 |cat>ch6f7s3

shreya@shreya-VirtualBox:~$ paste ch6f7s1 ch6f7s3> Ch6S7F2
```

5.Sort the file (Ch6S7F2) on login name without renaming it. Save the file.

```
shreya@shreya-VirtualBox:~$ sort +0 -1 Ch6S7F2 -o Ch6S7F2
```

6.Use the commands you have learned so far to reorganize the file Ch6S7F2 into six columns using the following format:
Note that you should divide the number of users by three to find out the number of lines in this new format. You should create three files and then paste them together

```
shreya@shreya-VirtualBox:~$ head -15 Ch6S7F2 | cat>f1
shreya@shreya-VirtualBox:~$ head -31 Ch6S7F2 | tail +16| cat>f2
shreya@shreya-VirtualBox:~$ head -47 Ch6S7F2 | tail +32| cat>f3
shreya@shreya-VirtualBox:~$ paste f1 f2 f3 >Ch6S7F3
shreya@shreya-VirtualBox:~$ cat Ch6S7F3
_apt  105   irc   39    shreya     1000
avahi 115   kernoops   116   speech-dispatcher     114
avahi-autoipd    109   list  38    sync  4
backup     34    lp    7     sys   3
bin   2     mail  8     syslog     104
colord     121   man   6     systemd-coredump 999
cups-pk-helper   113   messagebus 103   systemd-network  100
daemon     1     news  9     systemd-resolve  101
dnsmasq    112   nm-openvpn 118   systemd-timesync 102
games 5     nobody     65534 tcpdump    108
gdm   125   proxy 13    tss   106
geoclue    122   pulse 123   usbmux     110
gnats 41    root  0     uucp  10
gnome-initial-setup   124   rtkit 111   uuidd 107
hplip 119   saned 117   whoopsie   120
      shrey1110  1001  www-data   33
```

7.Quit the terminal.

```
shreya@shreya-VirtualBox:~$exit
```

Assignment 5: grep

Exercise 1

1. **Launch a terminal.**
2. **Use a command to create a file containing the following data. Call it a6-e1-f1. Use tabs to separate the fields Do not type the headings.**

ID	Name	Hours Worked	Hourly Pay
1425	Juan	18	14.25
4321	George	22	21.11
6781	Anna	44	16.77
1451	Ben	36	21.77
2277	Tuan	16	18.77

```
shreya@shreya-VirtualBox:~$ cat>a6-e1-f1
1425  Juan  18    14.25
4321  George  22  21.11
6781  Anna  44    16.77
1451  Ben   36    21.77
2277  Tuan    16  18.77
```

3. **Use a one-line command to display the hourly pay of Anna (only the last field).**

```
shreya@shreya-VirtualBox:~$ grep "Anna" a6-e1-f1 | cut -f4
16.77
```

4. **4.Use a one–line command to find the name of the**

employee with ID 1451.

```
shreya@shreya-VirtualBox:~$ grep "1451" a6-e1-f1 | cut -f2
Ben
```

5. **Use a one-line command to find the names of employees who worked more than 20 hours. Hint: You may first want to extract the second and third fields before applying one of the grep commands.**

```
shreya@shreya-VirtualBox:~$ cut -f2,3 a6-e1-f1 | grep "\<[2-9][1-
9].*" | cut -f1
George
Anna
Ben
```

6. **Use a one-line command to find the id and hours worked for employees who earn more than $20 per hour.**

```
shreya@shreya-VirtualBox:~$ cut -f1,3,4 a6-e1-f1 | grep "[2-9].\." |cut -
f1,2

4321  22

1451  36
```

7. **Use a one-line command to find the id, name, and hourly pay for employees who worked fewer than 10 hours**

```
shreya@shreya-VirtualBox:~$ grep "[0-9].?" a6-e1-f1 |cut -f1,2,4
```

Exercise 2

1. Launch a terminal.

2. Create the following file. Call it a6-e2-f1

```
shreya@shreya-VirtualBox:~$ cat>a6-e2-f1

 Psalm of Life

Tell me not, in mournful numbers,

 Life is but an empty dream!

For the soul is dead that slumbers,

 And things are not what they seem.


Life is real! Life is earnest!

 And the grave is not its goal;

Dust thou art, to dust returnest,

 Was not spoken of the soul.


Not enjoyment, and not sorrow,

 Is our destined end or way;

But to act, that each to-morrow

 Find us farther than to-day.


Art is long, and Time is fleeting,

 And our hearts, though stout and brave,
```

Still, like muffled drums, are beating
 Funeral marches to the grave.

In the world's broad field of battle,
 In the bivouac of Life,
Be not like dumb, driven cattle!
 Be a hero in the strife!

Trust no Future, howe'er pleasant!
 Let the dead Past bury its dead!
Act, act in the living Present!
 Heart within, and God o'erhead!

Lives of great men all remind us
 We can make our lives sublime,
And, departing, leave behind us
 Footprints on the sands of time;

Footprints, that perhaps another,
 Sailing o'er life's solemn main,
A forlorn and shipwrecked brother,
 Seeing, shall take heart again.

Let us, then, be up and doing,
 With a heart for any fate;
Still achieving, still pursuing,

Learn to labor and to wait.

a. Use a one-line command to copy file a6-e2-f1 without the blank lines (copy with- out nonblank lines). Call the new file a6-e2-f2.

```
shreya@shreya-VirtualBox:~$ grep "[^(^$)]" a6-e2-f1 |cat>a6-e2-f2

shreya@shreya-VirtualBox:~$ cat a6-e2-f2

 Psalm of Life

Tell me not, in mournful numbers,

 Life is but an empty dream!

For the soul is dead that slumbers,

 And things are not what they seem.

Life is real! Life is earnest!

 And the grave is not its goal;

Dust thou art, to dust returnest,

 Was not spoken of the soul.

Not enjoyment, and not sorrow,

 Is our destined end or way;

But to act, that each to-morrow

 Find us farther than to-day.

Art is long, and Time is fleeting,

 And our hearts, though stout and brave,

Still, like muffled drums, are beating

 Funeral marches to the grave.

In the world's broad field of battle,
```

```
  In the bivouac of Life,
Be not like dumb, driven cattle!
  Be a hero in the strife!
Trust no Future, howe'er pleasant!
  Let the dead Past bury its dead!
Act, act in the living Present!
  Heart within, and God o'erhead!
Lives of great men all remind us
  We can make our lives sublime,
And, departing, leave behind us
  Footprints on the sands of time;
Footprints, that perhaps another,
  Sailing o'er life's solemn main,
A forlorn and shipwrecked brother,
  Seeing, shall take heart again.
Let us, then, be up and doing,
  With a heart for any fate;
Still achieving, still pursuing,
  Learn to labor and to wait.
```

b. Use appropriate commands to create two new files out of a6-e2-f2. The first file, called a6-e2-f3, contains only the lines that are indented. The second file, called a6-e2-f4, contains the lines that are not indented.

```
shreya@shreya-VirtualBox:~$ grep "^[^A-Z].*" a6-e2-f2 |cat >a6-e2-f3

shreya@shreya-VirtualBox:~$ cat a6-e2-f3
  Psalm of Life
  Life is but an empty dream!
```

And things are not what they seem.

And the grave is not its goal;

Was not spoken of the soul.

Is our destined end or way;

Find us farther than to-day.

And our hearts, though stout and brave,

Funeral marches to the grave.

In the bivouac of Life,

Be a hero in the strife!

Let the dead Past bury its dead!

Heart within, and God o'erhead!

We can make our lives sublime,

Footprints on the sands of time;

Sailing o'er life's solemn main,

Seeing, shall take heart again.

With a heart for any fate;

Learn to labor and to wait.

```
shreya@shreya-VirtualBox:~$ grep "^[^A-Z].*" a6-e2-f2 |cat >a6-e2-f4
shreya@shreya-VirtualBox:~$ cat a6-e2-f4
Tell me not, in mournful numbers,
For the soul is dead that slumbers,
Life is real! Life is earnest!
Dust thou art, to dust returnest,
Not enjoyment, and not sorrow,
But to act, that each to-morrow
```

Art is long, and Time is fleeting,

Still, like muffled drums, are beating

In the world's broad field of battle,

Be not like dumb, driven cattle!

Trust no Future, howe'er pleasant!

Act, act in the living Present!

Lives of great men all remind us

And, departing, leave behind us

Footprints, that perhaps another,

A forlorn and shipwrecked brother,

Let us, then, be up and doing,

Still achieving, still pursuing,

c. Use a one-line command to create a new file out of a6-e3-f3. This file, which is called a6-e2-f5, contains the lines that are centered (equal leading and trailing spaces in each line).

```
shreya@shreya-VirtualBox:~$ cat -t a6-e2-f4 | grep -x '^[(^I)\|(
)*].*[(^I)\|( )*]$' |cat > a6-e2-f5
shar1234@MANI-PC:~$ cat a6-e2-f5
^ILife is but an empty dream!^I
^IAnd the grave is not its goal;^I
^IIs our destined end or way;^I
^IFind us farther than to-day.^I
```

d. Quit the terminal.

3. Exercise 3

a. Log into the system.

b. Create and save the following file. Do not type the headings. Call it a6-e3-f1

Department	Course	Session	Enrollment
CIS	15	1	45
CIS	54	1	20
BUS	34	2	20
ENG	11	2	89
CIS	45	1	38
MTH	35	1	56
MTH	35	2	41
PE	17	2	25
CIS	54	2	67

```
shreya@shreya-VirtualBox:~$ cat>a6-e3-f1
CIS   15    1     45
CIS   54    1     20
BUS   34    2     20
ENG   11    2     89
CIS   45    1     38
MTH   35    1     56
MTH   35    2     41
PE    17    2     25
CIS   54    2     67
```

a. Use a command to create a file of courses that have only one session. Call the file a6-e3-f2.

```
shreya@shreya-VirtualBox:~$ grep "\<1\>" a6-e3-f1 |cat >a6-e3-f2

shreya@shreya-VirtualBox:~$ cat a6-e3-f2

CIS   15    1     45

CIS   54    1     20

CIS   45    1     38

MTH   35    1     56
```

b. Use a command to create a file 0f courses offered in the CIS department. Call the file a6-e3-f3.

```
shreya@shreya-VirtualBox:~$ grep "CIS" a6-e3-f1 |cat >a6-e3-f3

shreya@shreya-VirtualBox:~$ cat a6-e3-f3

CIS   15    1     45

CIS   54    1     20

CIS   45    1     38

CIS   54    2     67
```

c. Use a command to create a file of courses that have fewer than 25 students. Call the file a6-e3-f4.

```
shreya@shreya-VirtualBox:~$ grep "[0-2][0-4]$" a6-e3-f1 |cat >a6-e3-f4

shreya@shreya-VirtualBox:~$ cat a6-e3-f4

CIS   54    1     20

BUS   34    2     20
```

d. Use a command to create a file of Courses that have between 25 and 30 students. Call the file a6-e3-f5.

```
shreya@shreya-VirtualBox:~$ grep "2[5-9]$\|30$" a6-e3-f1 |cat >a6-e3-f5

shreya@shreya-VirtualBox:~$ cat a6-e3-f5

PE   17    2     25
```

e. Quit the terminal.

Exercise 4

a. Log into the system.

b. Create and save the following file. Call it a6-e4-f1.

Great fleas have little fleas upon
their backs to bite 'em,
And little fleas have lesser fleas, and so
ad infinitum.
And the great fleas themselves, in turn,
have greater fleas to go on;
While these again have greater still, and
greater still, and so on.

```
shreya@shreya-VirtualBox:~$ cat>a6-e4-f1
Great fleas have little fleas
    upon their backs to bite 'em,
And little fleas have lesser fleas,
    and so ad infinitum.
And the great fleas themselves,
    in turn, have greater fleas to go on;
While these again have greater still,
    and greater still, and so on.
```

a. Use appropriate commands to create two files out of this file. The first one, a6-e4-f2, contains lines with no duplicated words. The second one, a6-e4-f3, has triplicated words.

```
shreya@shreya-VirtualBox:~$ grep -o '\(\<.*\>\).*\<\1\>' a6-e4-f1 | cut -
c 1-5 | cat >a6-e4-f3 |paste a6-e4-f1 a6-e4-f3 |cat >a6-e4-f3

shreya@shreya-VirtualBox:~$ paste  a6-e4-f12 a6-e4-f3

fleas have little fleas fleas

fleas have lesser fleas fleas

and greater still, and  and g
```

Exercise 5

Launch a terminal.

Create and save the following file. Call it a6-e5-f1.

*** Section**
A section heading starts with *.
**** Subsection**
A subsection heading starts with **.
We *emphasize* a phrase by delimiting it *.
***** Subsubsection**
A subsubsection starts with *.**
It is a great organization, indeed!

```
shreya@shreya-VirtualBox:~$ cat>a6-e5-f1
* Section
A section heading starts with *.
** Subsection
A subsection heading starts with **.
We *emphasize* a phrase by delimiting it *.
*** Subsubsection
A subsubsection starts with ***.
*It is a great organization, indeed!*


shreya@shreya-VirtualBox:~$ grep -x '\([*]\)*.*\1' a6-e5-f1
*It is a great organization, indeed!
```

b. Write a command that finds the lines that start with one and only one asterisk. The line may contain more asterisks, but it must start with one asterisk followed by a nonasterisk character.

```
shreya@shreya-VirtualBox:~$ grep '^\*[^*]' a6-e5-f1

* Section

*It is a great organization, indeed!*
```

Write a command that finds the lines that contain two asterisks separated by another single character.

```
shreya@shreya-VirtualBox:~$ grep '^\*.[^*]\*' a6-e5-f1
```

c. Write a command that finds lines with six or fewer asterisks.

```
shreya@shreya-VirtualBox:~$ grep -v '\(\*[^*]*\)\{7\}' a6-e5-f1
* Section

A section heading starts with *.

** Subsection

A subsection heading starts with **.

We *emphasize* a phrase by delimiting it *.

*** Subsubsection

A subsubsection starts with ***.

*It is a great organization, indeed!*
```

Assignment 7: sed and awk

Exercise 1

1. **Launch a terminal.**
2. **Create and save the following file. Note that the fields are sepa- rated by one or more spaces (randomly). The first field is the first name, the second field is the last name, and the third field is the age.**

   John Adams 55

   George Bull 77

   Anne Blue 99

   Janet Blue 67

   Ben Benjamin 78

   Ted White 32

```
shreya@shreya-VirtualBox:~$cat p7a
John Adams 55
George Bull 77
Anne Blue 99
Janet Blue 67
Ben Benjamin 78
Ted White 32
```

3. **Use a one-line sed command to reorganize the file using the comma/tab pattern shown in the following file. Note that the last name is before the first name, and there is only one space between the names and a space before the numbers. Use the same name for the new file.**

   Adams, John 55

   Bull, George 77

   Blue, Anne 99

   Blue, Janet 67

   Benjamin, Ben 78

   White, Ted 32

```
shreya@shreya-VirtualBox:~$sed -ie 's/\(.*\) \(.*\) \(.*\)/\2, \1 \3/g' p7a
shreya@shreya-VirtualBox:~$cat p7a
Adams, John 55
```

```
Bull, George 77
Blue, Anne 99
Blue, Janet 67
Benjamin, Ben 78
White, Ted 32
```

4. **Sort the file first according to the last name and then according to the age. Watch out for the comma after the last name. Use the same name for the new file.**

```
shreya@shreya-VirtualBox:~$sort -b +0 -1 +2n -3 p7a -o p7a
shreya@shreya-VirtualBox:~$cat p7a
Adams, John 55
Benjamin, Ben 78
Blue, Janet 67
Blue, Anne 99
Bull, George 77
White, Ted 32
```

5. **Use a sed script and a sed command to put a set of five asterisks at the beginning and end of each line that contains the pattern Blue.**

```
shreya@shreya-VirtualBox:~$cat > 7a1.sed
s/.*Blue.*/*****&*****/g
shreya@shreya-VirtualBox:~$sed -f 7a1.sed p7a
Adams, John 55
Benjamin, Ben 78
*****Blue, Janet 67*****
*****Blue, Anne 99*****
Bull, George 77
White, Ted 32
```

6. **Use the cat command to insert a line number at the beginning of each line in a file.**

```
shreya@shreya-VirtualBox:~$cat -n p7a
     1    Adams, John 55
     2    Benjamin, Ben 78
     3    Blue, Janet 67
     4    Blue, Anne 99
     5    Bull, George 77
     6    White, Ted 32
```

7. **Write a sed script and a sed command to split the file into three files. The first file, called f1, contains lines 2 and 3. The second file, called f2, contains lines 4 and 5. The third file,**

called f3, contains lines 1 and 6.

```
shreya@shreya-VirtualBox:~$cat > 71b.sed
1w f3
2,3w f1
4,5w f2
6w f3
shreya@shreya-VirtualBox:~$sed -n -f 71b.sed p7a
```

8. **Print all of the files created in this session and verify the output.**

```
shreya@shreya-VirtualBox:~$cat f1
Benjamin, Ben 78
Blue, Janet 67
shreya@shreya-VirtualBox:~$cat f2
Blue, Anne 99
Bull, George 77
shreya@shreya-VirtualBox:~$cat f3
Adams, John 55
White, Ted 32
```

9. **Quit the terminal.**

Exercise 2

10. **Launch a terminal.**

11. **Create the following file and call it a7-e4-f1. Each line in the file is an absolute pathname of a file.**

    bin/date bin/programs/cal

    usr/bin/date

    usr/report/file1

    usr/report/1etters/lett1

    /spool/mails

```
shreya@shreya-VirtualBox:~$cat > a7-e4-f1
/bin/date
/bin/programs/cal
/usr/bin/date
/usr/report/file1
/usr/report/letters/lett1
/spool/mails
```

12. **Write a sed script (a7-e4-f2) and a sed command to extract the lowest level direc- tory and the name of the file from the path (separated by spaces) and store it in a file called a7-e4-f3. The file should look like the following (directory then file}:**

    /bin date

    /bin/programs cal

    /usr/bin date

    /usr/report file1

    /usr/report/letters lett1

    /spool mails

```
shreya@shreya-VirtualBox:~$cat > a7-e4-f2.sed
s,/\(.*\)/\(.*\)$,/\1 \2,g
shreya@shreya-VirtualBox:~$sed -f a7-e4-f2.sed a7-e4-f1
/bin date
/bin/programs cal
/usr/bin date
/usr/report file1
/usr/report/letters lett1
/spool mails
```

13. **Quit the terminal.**

Exercise 3

14. **Launch a terminal.**

15. **Create the following file and call it a7-e6-f1. The file is a C program that multiplies two numbers. It contains some comments which begin with the two-character token**
/* and end with the two-character token */. In this program, comments can be on one line or can span more than a line

```
/* This program reads two integer numbers from the
    keyboard and prints their product.
    Written by:
    Date:


*/

/* Statements */

scanf ("%d", &number1); scanf
("%d", &number2); result =
number1 * number2; printf
("%d", result); return 0;
} /* main */
```

```
shreya@shreya-VirtualBox:~$cat > a7-e6-f1
/* This program reads two integer numbers from the
keyboard and prints their product.
Written by:
Date:
*/

/* Statements */

scanf ("%d", &number1);
scanf ("%d", &number2);
result = number1 * number2;
printf ("%d", result);
return 0;
}
/* main */
```

16. **Write a sed script a7-e6-f2.sed and a sed command to delete the comments from the file. Call the new file a7-e6-f3.**

```
shreya@shreya-VirtualBox:~$cat > a7-e6-f2.sed
/\/\*.*\*\/$/d
/\/\*/,/\*\/$/d
shreya@shreya-VirtualBox:~$sed -f a7-e6-f2.sed a7-e6-f1


scanf ("%d", &number1);
scanf ("%d", &number2);
result = number1 * number2;
printf ("%d", result);
return 0;
}
```

Exercise 4

1. Create the file named workers.txt with the following

First Name	Last Name	Rate	Hours
George	White	18.00	23
Mark	Red	18.10	20
Mary	Blue	10.89	25
Dan	Black	12.00	0
Susan	Green	18.00	40
Nora	Brown	17.20	46
Bruce	Purple	12.20	52
John	Gray	11.00	39
Bob	Gold	15.00	45
Steve	Silver	14.67	25

```
shreya@shreya-VirtualBox:~$cat > workers.txt
George   White   18.00   23
Mark     Red     18.10   20
Mary     Blue    10.89   25
Dan      Black   12.00   0
Susan    Green   18.00   40
Nora     Brown   17.20   46
Bruce    Purple  12.20   52
John     Gray    11.00   39
Bob      Gold    15.00   45
Steve    Silver  14.67   25
```

a. Write awk command to print the first and last name who did not work in the last week

```
shreya@shreya-VirtualBox:~$awk '$4==0 {print $1,"\t",$2}' workers.txt
```

```
Dan       Black
```

b. Write awk command to print the record of the employee whose rate is $15 or more

```
shreya@shreya-VirtualBox:~$awk '$3>=15 {print $0}' workers.txt
George   White   18.00   23
Mark     Red     18.10   20
Susan    Green   18.00   40
Nora     Brown   17.20   46
Bob      Gold    15.00   45
```

c. Write awk command to print the record whose first name is Mary

```
shreya@shreya-VirtualBox:~$awk '$1=="Mary" {print $0}' workers.txt
Mary     Blue    10.89   25
```

d. Write awk command to print the record of the employee whose rate between $1 and $18

```
shreya@shreya-VirtualBox:~$awk '$3>1 && $3<18 {print $0}' workers.txt
Mary     Blue    10.89   25
Dan      Black   12.00   0
Nora     Brown   17.20   46
Bruce    Purple  12.20   52
John     Gray    11.00   39
Bob      Gold    15.00   45
Steve    Silver  14.67   25
```

2. Create a file sales.txt with the following contents

```
Month      Sales
January    20
February   30
March      43
February   34
January    12
June       89
May        97
June       60
July       23
August          13
August          45
October    56
October    45
November   34
```

```
shreya@shreya-VirtualBox:~$cat > sales.txt
January          20
February 30
March            43
February 34
January          12
June             89
May              97
June             60
July             23
August           13
August           45
October          56
October          45
November 34
```

a. Write awk command to find total sales

```
shreya@shreya-VirtualBox:~$cat > 41.awk
BEGIN{
sales = 0;
}
{
sales+=$2;
}
END{
printf("Total sales : %d\n",sales);
}
shreya@shreya-VirtualBox:~$awk -f 41.awk sales.txt
Total sales : 601
```

b. Write awk script to find the total sales in every month

```
shreya@shreya-VirtualBox:~$cat > 42.awk
{
sales[$1]+=$2;
}
END{
for(i in sales) print(i, sales[i]);
}
shreya@shreya-VirtualBox:~$awk -f 42.awk sales.txt
June 149
October 101
March 43
August 58
July 23
May 97
```

```
February 64
January 32
November 34
```

c. Write awk script to find the months with no sales

```
shreya@shreya-VirtualBox:~$cat > 43.awk
BEGIN{
sales["January"]=0;
sales["February"]=0;
sales["March"]=0;
sales["April"]=0;
sales["May"]=0;
sales["June"]=0;
sales["July"]=0;
sales["August"]=0;
sales["September"]=0;
sales["October"]=0;
sales["November"]=0;
sales["December"]=0;
}
{
sales[$1]+=$2;
}
END{
for(i in sales) if(sales[i]==0) print i;
}
shreya@shreya-VirtualBox:~$awk -f 43.awk sales.txt
September
April
December
```

Exercise 8 - Simple Programs using shell scripts

Write a shell script that prints “Shell Scripting is Fun!” on the screen. Modify the shell script above to include a variable. The variable will hold the contents of the message “Shell Scripting is Fun!”

```
shreya@shreya-VirtualBox:~$ cat>8a.txt
#!/bin/bash
echo ""Shell Scripting is Fun!""
word="Shell Scripting is Fun!"
echo $word

shreya@shreya-VirtualBox:~$ bash 8a.txt
Shell Scripting is Fun!
Shell Scripting is Fun!
```

Print the values of the environment variables HOME, USER, SHELL and PATH with set, printenv and echo.

```
shreya@shreya-VirtualBox:~$ cat>8b.txt
echo "Using echo"
echo $HOME
echo $USER
echo $SHELL
echo $PATH
echo
echo "Using printenv"
printenv HOME
printenv USER
printenv SHELL
printenv PATH
echo
echo "Using set"
set | grep "HOME="
set | grep "USER=";
set | grep "SHELL="
set | grep "PATH=/"

shreya@shreya-VirtualBox:~$ bash 8b.txt
Using echo
/home/shreya
shreya
/bin/bash
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games:/snap/bin

Using printenv
```

```
/home/shreya
shreya
/bin/bash
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games:/snap/bin

Using set
HOME=/home/shreya
USER=shreya
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:
/usr/local/games:/snap/bin
```

Store the output of the command "hostname" in a variable. Display "This script is running on _." where "_" is the output of the "hostname" command.

```
shreya@shreya-VirtualBox:~$ cat>8c.txt
#!/bin/bash
s=`hostname`
echo "This script is running on $s"

shreya@shreya-VirtualBox:~$ bash 8c.txt
This script is running on shreya-VirtualBox
```

Write a shell script to calculate the net salary of an employee in a particular month considering various allowances (TA, DA, HRA) and deductions (INCOME TAX, PROVIDEND FUND) as:
a. TA=15 percent of basic salary
b. DA=2 percent of basic salary
c. HRA=10 percent of basic salary
d. INCOME TAX=5 percent of salary
e. PROVIDEND FUND=10 percent of salary

```
shreya@shreya-VirtualBox:~$ cat>8d.txt
#!/bin/bash
read -p "Enter basic salary" sal
TA=$((sal*15/100))
DA=$((sal*2/100))
HRA=$((sal*10/100))
INCOME_TAX=$((sal*5/100))
PROVIDEND_FUND=$((sal*10/100))
NETPAY=$((TA+DA+HRA+sal-INCOME_TAX-PROVIDEND_FUND))
echo "Netpay = $NETPAY"
shreya@shreya-VirtualBox:~$ bash 8d.txt
Enter basic salary7899
Netpay = 8846
```

5.In a town, the percentage of men is 52. Rest all are women. The percentage of total literacy is 48. If total percentage of literate men is 35 of the total population, WAP to find the total number of illiterate men and women. The population of the town is 80,000.

```
shreya@shreya-VirtualBox:~$ cat>8e.txt
#!/bin/bash
p=80000
m=$((p*52/100))
w=$((p*48/100))
lm=$((m*35/100))
im=$((m-lm))
l=$((p*48/100))
lw=$((l-lm))
iw=$((w-lw))
echo "Illiterate men : $im";
echo "Illiterate women : $iw";

shreya@shreya-VirtualBox:~$ bash 8e.txt
Illiterate men : 27040
Illiterate women : 14560
```

6. Temperature of a city in Fahrenheit degree is input through the keyboard WAP to convert this temperature into Centigrade degrees. Formula is c/100=f32/180 f=9/5*c+32

```
shreya@shreya-VirtualBox:~$ cat>8f.txt
#!/bin/bash
read -p "Enter temp in Farenheit" f
c=$(((f-32)*5/9))
echo "Temp in Celcius : $c";

shreya@shreya-VirtualBox:~$ bash 8f.txt
Enter temp in Farenheit99
Temp in Celcius : 37
```

7.The length & breadth of a rectangle and radius of a circle are input through the keyboard. Write a program to calculate the area & perimeter of the rectangle, and the area & circumference of the circle.

```
shreya@shreya-VirtualBox:~$ cat>8g.txt
#!/bin/bash
read -p "Enter length of rectangle" lr
read -p "Enter breadth of rectangle" br
```

```
ar=$((lr*br))
pr=$((2*(lr+br)))
echo "Area of rectangle : $ar";
echo "Perimeter of rectangle : $pr"
read -p "Enter radius of circle " rc
ac=$(echo "3.14*$rc*$rc"|bc)
cc=$(echo "3.14*2*$rc"|bc)
echo "Area of circle : $ac";
echo "Circumference of circle : $cc"

shreya@shreya-VirtualBox:~$ bash 8g.txt
Enter length of rectangle5
Enter breadth of rectangle4
Area of rectangle : 20
Perimeter of rectangle : 18
Enter radius of circle 10
Area of circle : 314.00
Circumference of circle : 62.80
```

Write a shell script that displays “man”,”bear”,”pig”,”dog”,”cat”,and “sheep” on the screen with each appearing on a separate line. Use special characters to display the filename, no of parameters, display the arguments each on one line, use appropriate command to dispaly the differences between $@, $*. Explain how about the status code of the script.

```
shreya@shreya-VirtualBox:~$ cat>7.8.txt

#!/bin/bash

for word in "$@"

do

      echo $word

done

echo



for word in "$*"

do

      echo $word
```

```
done

echo

echo "filename: $0"

echo "parameters: $#"

echo "status code: $?"
```

```
shreya@shreya-VirtualBox:~$ bash 7.8.txt man bear pig dog cat sheep

man

bear

pig

dog

cat

sheep


man bear pig dog cat sheep


filename: 7.8.txt

parameters: 6

status code: 0
```

Well executed unix code returns a 0 exit code.

Non zero exit code is returned when there is a failure in the code

Write a shell script that prompts the user for a name of a file or directory and reports if it is a regular file, a directory, or another type of file. Also perform an ls command against the file or directory with the long listing option.

```
shreya@shreya-VirtualBox:~$ cat>8.txt

#!/bin/bash
```

```
echo "Enter file path"
read FILE
if [ -f "$FILE" ]
      then
            echo "$FILE is a regular file"
elif [ -d "$FILE" ]
      then
            echo "$FILE is a directory"
else
      echo "$FILE is another type of file"
fi
ls -l $FILE

shreya@shreya-VirtualBox:~$ bash 8.txt
Enter file path
Pictures
Pictures is a directory
total 0
shreya@shreya-VirtualBox:~$ bash 8.txt
Enter file path
Desktop
Desktop is a directory
total 4
shreya@shreya-VirtualBox:~$ bash 8.txt
Enter file path
ex7.txt
```

```
ex7.txt is a regular file

-rw-rw-r-- 1 shreya shreya 8192 Oct 16 14:16 ex7.txt

shreya@shreya-VirtualBox:~$ bash 8.txt

Enter file path

yesssssssssss

yesssssssssss is another type of file

ls: cannot access 'yesssssssssss': No such file or directory
```

10. Modify the previous script to that it accepts the file or directory name as an argument instead of prompting the user to enter it.

```
shreya@shreya-VirtualBox:~$ cat>8.2.txt

#!/bin/bash

if [ -f "$@" ]

      then

            echo "$@ is a regular file"

elif [ -d "$@" ]

      then

            echo "$@ is a directory"

else

      echo "$@ is another file"

fi

ls -l "$@"


shreya@shreya-VirtualBox:~$ bash 8.2.txt Pictures

Pictures is a directory
```

```
total 0

shreya@shreya-VirtualBox:~$ bash 8.2.txt s1.txt

s1.txt is a regular file

-rw-rw-r-- 1 shreya shreya 13 Sep 28 08:45 s1.txt

shreya@shreya-VirtualBox:~$ bash 8.2.txt yesssssssssssssssssss

yesssssssssssssssssss is another file

ls: cannot access 'yesssssssssssssssssss': No such file or directory
```

Modify the previous script to accept an unlimited number of files and directories as arguments.

```
shreya@shreya-VirtualBox:~$ cat > 8.3.txt

#!/bin/bash

for file in "$@"

      do

            if [ -f "$file" ]

                  then

                        echo "$file is a regular file"

            elif [ -d "$file" ]

                  then

                        echo "$file is a directory"

            else

                  echo "$file is another type of file"

            fi

            ls -l $file


shreya@shreya-VirtualBox:~$ bash 8.3.txt yessss hi Pictures

8.3.txt: line 14: syntax error: unexpected end of file

shreya@shreya-VirtualBox:~$ cat >> 8.3.txt
```

```
done

shreya@shreya-VirtualBox:~$ bash 8.3.txt yessss hi Pictures

yessss is another type of file

ls: cannot access 'yessss': No such file or directory

hi is a directory

total 0

Pictures is a directory

total 0
```

Write a script program to get the marks of 5 different subjects and find the total. Print the total.(use brace expansion)

```
shreya@shreya-VirtualBox:~$ cat > 12.txt

total=0

for i in {0..4}

      do

            echo "Enter marks $(( i+1 ))"

            read mark

            (( total=total+mark ))

done

echo "Total marks: $total"

shreya@shreya-VirtualBox:~$ bash 12.txt

Enter marks 1

67

Enter marks 2

99

Enter marks 3

88
```

```
Enter marks 4

99

Enter marks 5

99

Total marks: 452
```

Write a shell script to display the current date and cut down the month of the date and store it in the file date.txt. Use ` ` in the command to store the content in the file and display the file. Also create an alias for the entire command and try to execute the command.

```
shreya@shreya-VirtualBox:~$ cat>13.txt

#!/bin/bash

echo "Todays date without month is $(echo $('date')|cut -d " " -f 1,2,4-7)">date.txt

shopt -s expand_aliases

alias datecomm='echo "Todays date is $(echo $('date')|cut -d " " -f 1,2,4-7)"'

datecomm

shreya@shreya-VirtualBox:~$ bash 13.txt

Todays date is Saturday 17 2020 05:01:50 PM IST

shreya@shreya-VirtualBox:~$ cat date.txt

Todays date without month is Saturday 17 2020 05:01:50 PM IST
```

Create the following files and change the permissions specified
File1 701

File2 400
File3 300
File4 676
File5 045
File6 177
File7 234
File8 507

Write a shell script to find the number of readable, writable and executable files

```
shreya@shreya-VirtualBox:~$ cat>14.txt

#!/bin/bash

for file in "$@"

do

      touch $file

done

chmod u=rwx,g-r-w,o-r-w+x File1

chmod u=r,g-r-w-x,o-r-w-x File2

chmod u=wx,g-r-w-x,o-r-w-x File3

chmod u=rw,g=rwx,o=rw File4

chmod u-r-w-x,g=r,o=rw File5

chmod u=x,g=rwx,o=rwx File6

chmod u=w,g=wx,o=r File7

chmod u=rx,g-r-w-x,o=rwx File8

ec=0

rc=0

rc=0

for file in "$@"

do

      if test -w "$file"
```

```
            then
                  (( wc=wc+1 ))
      fi
      if test -r "$file"
            then
                  (( rc=rc+1 ))
      fi
      if test -x "$file"
            then
                  (( ec=ec+1 ))
      fi
done
echo "Readable files: $rc"
echo "Writable files: $wc"
echo "Executable files: $ec"
shreya@shreya-VirtualBox:~$ bash 14.txt File1 File2 File3 File4 File5 File6
File7 File8
Readable files: 4
Writable files: 4
Executable files: 4
```

Create a script for string to match patterns at the prefix and suffix
v0=1234abczip1234abc
pat=1*4
a) Replace the first occurence of the string “abc” with “ABC”
b) Replace the string “abc” with “ABC” globally.
c) Remove pat present at the prefix
d) Remove the longest possible pat from the string using (%%)

```
shreya@shreya-VirtualBox:~$ cat>7.15.txt
#!/bin/bash
v0="1234abczip1234abc"
echo ${v0/abc/ABC}
echo
echo ${v0//abc/ABC}
echo ${string#$pat}

shreya@shreya-VirtualBox:~$bash 7.15.txt
1234ABCzip1234abc
1234ABCzip1234ABC
```

Exercise 8
Control Flow , Loops , Functions and Arrays

Exercise 1

1. **Write a shell script that prints 5 command line arguments. What happens if we pass fewer than 5 arguments?**
2. **Change the value of a positional parameter. Did you succeed?**

```
shreya@shreya-VirtualBox:~$ cat>9.1.txt
#!/bin/bash

for i in {1..5}
do
        echo ${!i}
done
set bye world
echo "After modifying the positional parameters"
echo "$*"

shreya@shreya-VirtualBox:~$ bash 9.1.txt hi hello world unix lab
hi
hello
world
unix
lab
After modifying the positional parameters
bye world
```

Exercise 2

1. **Write shell script to read a text file name and count the number of lines using function. Pass the file name as an argument to the function. Return the number of lines and print it**

```
shreya@shreya-VirtualBox:~$ cat>9.2.txt
#!/bin/bash

function countwords()
{
     count=$(cat $1|wc -l)
}
echo "Enter filename: "
read filename
countwords $filename
echo "Number of lines: $count"
```

```
shreya@shreya-VirtualBox:~$ bash 9.2.txt
Enter filename:
9.1.txt
Number of lines: 9
```

2. **Write a shell script to count the number of occurrences of given word in the file. (Note: File name and word to be passed as an argument to the script).**

```
shreya@shreya-VirtualBox:~$ cat>9.3.txt
#!/bin/bash
occur()
{
        count=$(cat $filename|grep -c $word)
}
echo "Enter filename: "
read filename
if [ -f $filename ];then
        echo "Enter word: "
        read word
        occur $filename,$word
        echo "Number of occurences of the word: $count"
else
        echo "File not found"
fi

shreya@shreya-VirtualBox:~$ bash 9.3.txt
Enter filename:
9.2.txt
Enter word:
echo
Number of occurences of the word: 2
```

3. **Anna University converts the marks in an exam to letter grades according to the following table. Write a shell script to translate the marks of a student in a semester into letter grades.**

```
shreya@shreya-VirtualBox:~$ cat>9.4.txt
gradecalculator()
{
        for i in "${marks[@]}"
                do
                        if [ $i -ge 91 ]
                                then
                                        grade=“S”
                        elif [ $i -ge 81 ]
                                then
```

```
                                        grade="A"
                                elif [ $i -ge 71 ]
                                        then
                                                grade="B"
                                elif [ $i -ge 61 ]
                                        then
                                                grade="C"
                                elif [ $i -ge 51 ]
                                        then
                                                grade="D"
                                else
                                        grade="E"
                                fi
                        echo "$i:$grade"
                done
        }

        echo "Enter marks"
        read -a marks
        gradecalculator $marks
```

```
shreya@shreya-VirtualBox:~$ bash 9.4.txt
Enter marks
99 89 77 65 43 22 12
99:S
89:A
77:B
65:C
43:E
22:E
12:E
```

Exercise 3 (loops)

1. **Write a program to generate all combinations of digits 1, 2 and 3 to form different numbers using for loops.**

```
shreya@shreya-VirtualBox:~$ cat>9.5.txt

#!/bin/bash
for i in {1..3}
        do
                for j in {1..3}
                        do
                                for k in {1..3}
                                        do
                                                echo $i$j$k
                                done
                done
done
```

```
shreya@shreya-VirtualBox:~$ bash 9.5.txt
111
112
113
121
122
123
131
132
133
211
212
213
221
222
223
231
232
233
311
312
313
321
322
323
331
332
333
```

2. Use seq with for statement to print the multiplication table.

```
shreya@shreya-VirtualBox:~$ cat 9.6.txt

#!/bin/bash

for i in `seq 5 9`
        do
                for j in {1..10}
                        do
                                echo "$i*$j=$(( $i*$j ))"
                done
done

shreya@shreya-VirtualBox:~$ bash 9.6.txt
Tables of 5

5*1=5
5*2=10
```

5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50

Tables of 6

6*1=6
6*2=12
6*3=18
6*4=24
6*5=30
6*6=36
6*7=42
6*8=48
6*9=54
6*10=60

Tables of 7

7*1=7
7*2=14
7*3=21
7*4=28
7*5=35
7*6=42
7*7=49
7*8=56
7*9=63
7*10=70

Tables of 8

8*1=8
8*2=16
8*3=24
8*4=32
8*5=40
8*6=48
8*7=56
8*8=64
8*9=72
8*10=80

Tables of 9

```
9*1=9
9*2=18
9*3=27
9*4=36
9*5=45
9*6=54
9*7=63
9*8=72
9*9=81
9*10=90
```

3. **Write a shell script to check whether a given string is a palindrome or not**

```
shreya@shreya-VirtualBox:~$ cat>9.7.txt
#!/bin/bash
read -p "Enter string: " word
let len=${#word}
for i in $(seq 1 $(( len/2 )))
do
        c1=`echo $word|cut -c$i`
        c2=`echo $word|cut -c$(( $len-$i+1 ))`
        if [[ $c1 != $c2 ]] ; then
                echo "Not palindrome"
                exit 0;
        fi
done
echo "Palindrome"

shreya@shreya-VirtualBox:~$ bash 9.7.txt
Enter string: hello
Not palindrome

shreya@shreya-VirtualBox:~$ bash 9.7.txt
Enter string: malayalam
Palindrome
```

4. **Write a shell script to compute ‘m’ to the power of a positive integer ‘n’, i.e. mn (while loop)**

```
shreya@shreya-VirtualBox:~$ cat>9.8.txt
#!/bin/bash
a=1
ans=1
read -p "Enter base number: " m
read -p "Enter power: " n
if [ $n -gt 0 ] ; then
        while [ $a -le $n ]
        do
```

```
                (( ans=$ans*$m ))
                let n+=1
        done
        echo "Exponent is: "$ans
else
        echo "Enter a positive power "
fi
```

```
shreya@shreya-VirtualBox:~$ bash 9.8.txt
Enter base number: 4
Enter power: 3
Exponent is: 64

shreya@shreya-VirtualBox:~$ bash 9.8.txt
Enter base number: 5
Enter power: 6
Exponent is: 15625
```

5. **Write a script that attempts to copy a file to a directory and, if it fails, waits 5 seconds and then tries again continuing until it succeeds. (Use Until statement)**

```
shreya@shreya-VirtualBox:~$ cat>9.9.txt
#!/bin/bash
read -p "Enter file path to copy: " filename
read -p "Enter destination folder path: " path
until ( cp $filename $path )
do
        echo "Waiting to copy"
        sleep 5
done

shreya@shreya-VirtualBox:~$ bash 9.9.txt
Enter file path to copy: 9.8.txt
Enter destination folder path: assignments/UNIX
```

Exercise 4. (conditional Statements)

1. **If cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred.**

```
shreya@shreya-VirtualBox:~$ cat>9.10.txt
#!/bin/bash
read -p "Enter cost price : " cp
read -p "Enter selling price : " sp
if [ $sp -gt $cp ]
        then
                p=$((sp-cp))
                echo "Profit : $p"
elif [ $sp -lt $cp ]
        then
                l=$((cp-sp))
                echo "Loss : $l"
else
        echo "Neither profit nor loss !"
fi

shreya@shreya-VirtualBox:~$ bash 9.10.txt
Enter cost price : 9700
Enter selling price : 8000
Loss : 1700

shreya@shreya-VirtualBox:~$ bash 9.10.txt
Enter cost price : 500
Enter selling price : 980
Profit : 480

shreya@shreya-VirtualBox:~$ bash 9.10.txt
Enter cost price : 300
Enter selling price : 300
Neither profit nor loss !
```

2. Write a shell script to validate password strength. Here are a few assumptions for the password string.
•Length – minimum of 8 characters.
•Contain both alphabet and number.
•Include both the small and capital case letters.
If the password doesn't satisfy with any of the above conditions, then the script should print it as a "Weak Password"

```
shreya@shreya-VirtualBox:~$ cat>9.11.txt
#!/bin/bash
read -p "Enter password : " pw
c=0
if [ ${#pw} -ge 8 ]
        then
                ((c=c+1))
fi
if [[ $pw =~ [A-Z] ]]
        then
                ((c=c+1))
fi
if [[ $pw =~ [a-z] ]]
        then
                ((c=c+1))
fi
if [[ $pw =~ [0-9] ]]
        then
                ((c=c+1))
fi
if [ $c -lt 4 ]
        then
                echo "$pw is a weak password !"
else
        echo "$pw is a strong password "
fi
shreya@shreya-VirtualBox:~$ bash 9.11.txt
Enter password : UnixLab
UnixLab is a weak password !
shreya@shreya-VirtualBox:~$ bash 9.11.txt
Enter password : UnixLab01
UnixLab01 is a strong password
```

3. Write a script that prints essentially the same information as ls l a but in a more userfriendly way.
(a) file exists or not
(b) regular file?
(c) directory?
(d) readable?
(e) writable?
(f) executable?
(g) owner

```
shreya@shreya-VirtualBox:~$ cat>9.12.txt
#!/bin/bash
read -p "Enter filename : " file
```

```
if [[ -f "$file" || -d "$file" ]]
        then
                echo "$file exists"
                if [ -f "$file" ]
                        then
                                echo "$file is a regular file"
                elif [ -d "$file" ]
                        then
                                echo "$file is a directory"
                else
                        echo "$file is another type of file"
                fi
                p=0
                if test -r "$file"
                        then
                                ((p=p+1))
                                echo "$file is readable"
                fi
                if test -w "$file"
                        then
                                ((p=p+1))
                                echo "$file is writable"
                fi
                if test -x "$file"
                        then
                                ((p=p+1))
                                echo "$file is executable"
                fi
                if [ $p -eq 0 ]
                        then
                                echo "$file has no permissions"
                fi
                echo "Owner of $file : $(ls -l|cut -d " " -f 3|head -
2|tail +1)"
else
        echo "$file does not exist !"
fi

shreya@shreya-VirtualBox:~$ bash 9.12.txt
Enter filename : hello.txt
hello.txt exists
hello.txt is a regular file
hello.txt is readable
hello.txt is writable
Owner of hello.txt :
shreya
shreya@shreya-VirtualBox:~$ bash 9.12.txt
Enter filename : Documents
Documents exists
```

```
Documents is a directory
Documents is readable
Documents is writable
Documents is executable
Owner of Documents :
Shreya

shreya@shreya-VirtualBox:~$ bash 9.12.txt
Enter filename : unix.txt
unix.txt does not exist !
```

Print suitable messages. Rewrite the above script as a shell function finfo and call the function with a filename.

```
shreya@shreya-VirtualBox:~$ cat>9.13.txt
#!/bin/bash
finfo(){
if [[ -f "$1" || -d "$1" ]]
        then
                echo "$1 exists"
                if [ -f "$1" ]
                        then
                                echo "$1 is a regular file"
                elif [ -d "$1" ]
                        then
                                echo "$1 is a directory"
                else
                        echo "$1 is another type of file"
                fi
                p=0
                if test -r "$1"
                        then
                                ((p=p+1))
                                echo "$1 is readable"
                fi
                if test -w "$1"
                        then
                                ((p=p+1))
                                echo "$1 is writable"
                fi
                if test -x "$1"
                        then
                                ((p=p+1))
                                echo "$1 is executable"
                fi
                if [ $p -eq 0 ]
                        then
```

```
                                        echo "$1 has no permissions"
                fi
                echo "Owner of $1 : $(ls -l|cut -d " " -f 3|head -2|tail
+1)"
else
        echo "$1 does not exist !"
fi
}
finfo $1

shreya@shreya-VirtualBox:~$ bash 9.13.txt
 does not exist !
shreya@shreya-VirtualBox:~$ bash 9.13.txt hello.txt
hello.txt exists
hello.txt is a regular file
hello.txt is readable
hello.txt is writable
Owner of hello.txt :
shreya
shreya@shreya-VirtualBox:~$ bash 9.13.txt Documents
Documents exists
Documents is a directory
Documents is readable
Documents is writable
Documents is executable
Owner of Documents :
shreya
shreya@shreya-VirtualBox:~$ bash 9.13.txt unix.txt
unix.txt does not exist !
```

Exercise 5

1. Develop an interactive script to maintain a database of employees. The database is in the format
employee_name rate_per_hour hours_worked
The script should allow users to
1. List the records
2. Search for an employee
3. Modify the hours_worked of an employee whose existing hours_worked is equal to 0.
4. Delete an employee
5. Quit

```
shreya@shreya-VirtualBox:~$ cat>9.14.txt
#!/bin/bash
op=1
while [ $op -eq 1 ]
        do
                echo "Menu : "
                echo "1.List the records"
                echo "2.Search for an employee"
                echo "3.Modify hours worked"
                echo "4.Delete an employee"
                echo "5.Quit"
                read -p "Enter choice : " ch
                echo
                if [ $ch -eq 1 ]
                   then
                           cat data
                   echo
                elif [ $ch -eq 2 ]
                   then
                           read -p "Enter name : " n
                           s=`grep $n data|cut -f 1`
                           if [[ $s == $n ]]
                                   then
                                           echo "$(grep $n data)"
                           else
                                   echo "$n details not available"
                           fi
                           echo
                elif [ $ch -eq 3 ]
                   then
                           if [ -f test ]
                                   then
                                           rm test
                           fi
                           IFS=$"\n"
                           for l in $(cat data)
                                   do
                                   IFS=$"\t"
                                   read -a arr <<< "$l"
                                   n=${arr[2]}
                                   if [[ $n == "0" ]]
                                           then
                                                   n="5"
                                   fi
                                   echo "${arr[0]} ${arr[1]}
$n">>test
                                   done
                           mv test data
                           echo "After modification..."
```

```
                        cat data
                elif [ $ch -eq 4 ]
                        then
                                if [ -f test ]
                                        then
                                                rm test
                                fi
                                read -p "Enter name : " n
                                s=`grep $n data|cut -f 1`
                                if [[ $s == $n ]]
                                        then
                                                echo "$(grep -v $n
data)">>test
                                                mv test data
                                                echo "After deletion..."
                                                cat data
                                else
                                        echo "$n details not available"
                                fi
                                echo
                elif [ $ch -eq 5 ]
                        then
                                exit
                else
                        echo "Invalid option !"
                        echo
                fi
                read -p "Enter 1 to continue : " op
                echo
        done

shreya@shreya-VirtualBox:~$ cat>data
Beth    4.00    0
Dan     3.75    0
Kathy   4.00    10
Mark    5.00    20
Mary    5.50    22
Susie   4.25    18
shreya@shreya-VirtualBox:~$ bash 9.14.txt
Menu :
1.List the records
2.Search for an employee
3.Modify hours worked
4.Delete an employee
5.Quit
Enter choice : 1

Beth    4.00    0
Dan     3.75    0
```

```
Kathy   4.00    10
Mark    5.00    20
Mary    5.50    22
Susie   4.25    18

Enter 1 to continue : 1

Menu :
1.List the records
2.Search for an employee
3.Modify hours worked
4.Delete an employee
5.Quit
Enter choice : 2

Enter name : Kathy
Kathy   4.00    10

Enter 1 to continue : 1

Menu :
1.List the records
2.Search for an employee
3.Modify hours worked
4.Delete an employee
5.Quit
Enter choice : 2

Enter name : Shreya
Shreya details not available

Enter 1 to continue : 1

Menu :
1.List the records
2.Search for an employee
3.Modify hours worked
4.Delete an employee
5.Quit
Enter choice : 3

After modification...
Beth    4.00    5
Dan     3.75    5
Kathy   4.00    10
Mark    5.00    20
Mary    5.50    22
Susie   4.25    18
Enter 1 to continue : 1
```

```
Menu :
1.List the records
2.Search for an employee
3.Modify hours worked
4.Delete an employee
5.Quit
Enter choice : 4

Enter name : Mary
After deletion...
Beth    4.00    5
Dan     3.75    5
Kathy   4.00    10
Mark    5.00    20
Susie   4.25    18

Enter 1 to continue : 1

Menu :
1.List the records
2.Search for an employee
3.Modify hours worked
4.Delete an employee
5.Quit
Enter choice : 4

Enter name : Shreya
Shreya details not available

Enter 1 to continue : 1

Menu :
1.List the records
2.Search for an employee
3.Modify hours worked
4.Delete an employee
5.Quit
Enter choice : 5
```

2. Create an array by assignment of prices for five different fruits with fruit name as
key and price as value.
a. Display the all the key.
b. Display the values.
c. Display the key value pair.
d. Remove the third fruit.
e. Add one new fruit.

f. Calculate the total cost of all fruits and display the amount.
g. Delete the all items and display

```
shreya@shreya-VirtualBox:~$ cat>9.15.txt
#!/bin/bash
declare -A arr
arr[orange]=20
arr[apple]=10
arr[grapes]=15
arr[kiwi]=40
arr[banana]=25
echo "Displaying keys..."
for key in "${!arr[@]}"
        do
                echo "$key"
        done
echo
echo "Displaying values..."
for val in "${arr[@]}"
        do
                echo "$val"
        done
echo
echo "Displaying key and value pairs..."
for key in "${!arr[@]}"
        do
                echo "Key : $key Value : ${arr[$key]}"
        done
echo
i=0
for key in "${!arr[@]}"
        do
                ((i=i+1))
                if [ $i -eq 3 ]
                        then
                                echo "Third fruit : $key"
                                unset arr[$key]
                fi
        done
echo "After deletion of third fruit..."
for key in "${!arr[@]}"
        do
                echo "Key : $key Value : ${arr[$key]}"
        done
echo
arr+=([papaya]=50)
echo "After addition..."
for key in "${!arr[@]}"
```

```
        do
                echo "Key : $key Value : ${arr[$key]}"
        done
echo
total=0
for val in "${arr[@]}"
        do
                total=$((total+val))
        done
echo "Total price : $total"
echo
unset arr
echo "After deletion of array..."
echo "${arr[@]}"
shreya@shreya-VirtualBox:~$ bash 9.15.txt
Displaying keys...
grapes
orange
banana
apple
kiwi

Displaying values...
15
20
25
10
40

Displaying key and value pairs...
Key : grapes Value : 15
Key : orange Value : 20
Key : banana Value : 25
Key : apple Value : 10
Key : kiwi Value : 40

Third fruit : banana
After deletion of third fruit...
Key : grapes Value : 15
Key : orange Value : 20
Key : apple Value : 10
Key : kiwi Value : 40

After addition...
Key : grapes Value : 15
Key : orange Value : 20
Key : papaya Value : 50
Key : apple Value : 10
Key : kiwi Value : 40
```

```
Total price : 135

After deletion of array...
```

Excerise 6.
1. Write a function that allows the user to select a directory from the list of directories.
Move the selected directory to the first position of the list. (Using select statement).

```
shreya@shreya-VirtualBox:~$ cat>9.16.txt
#!/bin/bash
arr=(Desktop Documents Pictures)
select a in ${arr[@]}
        do
                if [ $REPLY -le 3 ]
                        then
                                i=1
                                echo "User choice = $REPLY Value = $a"
                                echo "Modified list : "
                                echo "$i) $a"
                                ((i=i+1))
                                for x in ${arr[@]}
                                        do
                                                if [[ $x != $a ]]
                                                        then
                                                                echo "$i) $x"
                                                                ((i=i+1))
                                                fi
                                        done
                else
                        echo "Invalid option !"
                fi
        break
        done
shreya@shreya-VirtualBox:~$ bash 9.16.txt
1) Desktop
2) Documents
3) Pictures
#? 1
User choice = 1 Value = Desktop
Modified list :
1) Desktop
2) Documents
3) Pictures
shreya@shreya-VirtualBox:~$ bash 9.16.txt
1) Desktop
```

```
2) Documents
3) Pictures
#? 2
User choice = 2 Value = Documents
Modified list :
1) Documents
2) Desktop
3) Pictures
shreya@shreya-VirtualBox:~$ bash 9.16.txt
1) Desktop
2) Documents
3) Pictures
#? 3
User choice = 3 Value = Pictures
Modified list :
1) Pictures
2) Desktop
3) Documents
shreya@shreya-VirtualBox:~$ bash 9.16.txt
1) Desktop
2) Documents
3) Pictures
#? 4
Invalid option !
```

2. Write a shell script to translate the contents of a file into Upper case, Lower case, title case and print not valid case when invalid argument passed where file name is entered through command line.(use select case)

```
shreya@shreya-VirtualBox:~$ cat>9.17.txt
#!/bin/bash
if [ -f "$1" ]
        then
                select ch in Title_case Upper_case Lower_Case
                        do
                                if [ $REPLY -le 3 ]
                                        then
                                                case $REPLY in
                                                        1) cat $1|tr
[:upper:] [:lower:]|sed -e “s/\<./\U&/g”
                                                        break;;
                                                        2) cat $1|tr
[:lower:] [:upper:]
                                                        break;;
```

```
                                                        3) cat $1|tr
[:upper:] [:lower:]
                                                        break;;
                                                esac
                                else
                                        echo "Invalid option !"
                                fi
                        break
                        done
else
        echo "Invalid argument! File $1 doesn”t exist!"
fi
```

```
shreya@shreya-VirtualBox:~$ cat hello.txt
hello world
bye
shreya@shreya-VirtualBox:~$ bash 9.17.txt hello.txt
1) Title_case
2) Upper_case
3) Lower_Case
#? 1
Hello World
Bye
shreya@shreya-VirtualBox:~$ bash 9.17.txt hello.txt
1) Title_case
2) Upper_case
3) Lower_Case
#? 2
HELLO WORLD
BYE
shreya@shreya-VirtualBox:~$ bash 9.17.txt hello.txt
1) Title_case
2) Upper_case
3) Lower_Case
#? 3
hello world
bye
shreya@shreya-VirtualBox:~$ bash 9.17.txt hello.txt
1) Title_case
2) Upper_case
3) Lower_Case
#? 4
Invalid option !
```

Assignment – 9 Bash Script

1. **Write a function mkcd() which would create all the directories present in the path supplied to it as argument and change over to the last directory in the path. Thus $mkcd d1/d2/d3 should create three directories and change the present working directory to d3**

```
shreya@shreya-VirtualBox:~$ cat>10.1.txt
#!/bin/bash

mkcd()
{
        readarray -d/ -t dir <<< "$1"
        for i in "${dir[@]}"
                do
                        mkdir $i
                done
        cd $i
        pwd
}
mkcd "$*"

shreya@shreya-VirtualBox:~$ bash 10.1.txt d1/d2/d3
/home/shreya/d3
```

2. **Write a program to display the total no of line, no of characters and no of words present in a file that has been passed as argument. Also display the count of total number of lines across a set of files.**

```
shreya@shreya-VirtualBox:~$ cat>10.2.txt
#!/bin/bash

let n=0
for file in "$@"
do
        echo "Filename: $file"
        echo "Line count: $(wc -l $file |cut -d " " -f 1)"
        echo "Word count: $(wc -w $file |cut -d " " -f 1)"
        echo "Character count: $(wc -c $file |cut -d " " -f 1)"
        (( n=n+`wc -l $file |cut -d " " -f 1` ))
        echo
done
echo "Total number of lines : $n"
```

```
shreya@shreya-VirtualBox:~$ bash 10.2.txt 9.1.txt 9.3.txt 9.5.txt
Filename: 9.1.txt
Line count: 9
Word count: 20
Character count: 123

Filename: 9.3.txt
Line count: 15
Word count: 41
Character count: 262

Filename: 9.5.txt
Line count: 13
Word count: 21
Character count: 125

Total number of lines : 37
```

3. **Write a menu based program to copy a given file, to remove the specified file and to move a file.**

```
shreya@shreya-VirtualBox:~$ cat 10.3.txt
#!/bin/bash

echo "1.Copy 2.Move 3.Remove"
echo
read -p "Enter option: " op
if [ $op -eq 1 ] ; then
        read -p "Enter source: " source
        read -p "Enter destination: " destination
        if [ -f "$source" ]
                then
                        `cp $source $destination`
                        echo "Contents after copying: "
                        echo "`cat $destination`"
        else
                echo "File doesnt exist"
        fi
        echo
elif [ $op -eq 2 ] ; then
        read -p "Enter source: " source
        read -p "Enter destination: " destination
        echo "File path before moving: `realpath -e $source`"
        if [ -f "$source" ] ; then
                `mv $source $destination`
                cd $destination
                echo "Path after moving: "
                echo "`realpath -e $source`"
```

```
        else
                echo "File doesnt exist"
        fi
        echo
elif [ $op -eq 3 ] ; then
        read -p "Enter file: " source
        if [ -f "$source" ]
                then
                        `rm $source`
                        echo "Contents after removing: "
                        echo "`cat $source`"
        else
                echo "File doesnt exist"
        fi
        echo

else
        echo "Invalid"
fi
```

```
shreya@shreya-VirtualBox:~$ bash 10.3.txt
1.Copy 2.Move 3.Remove

Enter option: 1
Enter source: a1.txt
Enter destination: a2.txt
Contents after copying:
hello this is test 1
bye

shreya@shreya-VirtualBox:~$ bash 10.3.txt
1.Copy 2.Move 3.Remove

Enter option: 1
Enter source: a1
Enter destination: a2.txt
File doesnt exist


shreya@shreya-VirtualBox:~$ bash 10.3.txt
1.Copy 2.Move 3.Remove

Enter option: 2
Enter source: a2.txt
Enter destination: Pictures
File path before moving: /home/shreya/a2.txt
Path after moving:
/home/shreya/Pictures/a2.txt
```

```
shreya@shreya-VirtualBox:~$ bash 10.3.txt
1.Copy 2.Move 3.Remove

Enter option: 2
Enter source: a2
Enter destination: Pictures
realpath: a2: No such file or directory
File path before moving:
File doesnt exist

shreya@shreya-VirtualBox:~$ bash 10.3.txt
1.Copy 2.Move 3.Remove

Enter option: 3
Enter file: a2.txt
File doesnt exist


shreya@shreya-VirtualBox:~$ bash 10.3.txt
1.Copy 2.Move 3.Remove

Enter option: 3
Enter file: a1.txt
Contents after removing:
cat: a1.txt: No such file or directory
```

4. **Write a program for reversing the digits by obtaining the value through command line arguments**

```
shreya@shreya-VirtualBox:~$ cat 10.4.txt
#!/bin/bash
num=0
rev=0
num="$1"
echo $num
while [ $num -gt 0 ]
do
        (( rem=num%10 ))
        (( rev=10*rev+rem ))
        (( num=num/10 ))
done
echo "Reversed number is : $rev"

shreya@shreya-VirtualBox:~$ bash 10.4.txt 123
123
Reversed number is : 321
```

5. **Write a script to check whether the files supplied as arguments are present or not. If any of the files do not exist then if a sub-directory called mydir exists in the current directory. If it didn't exists then create the directory with files supplied as arguments. If mydir already exists it should report with number of files that are currently present in mydir.**

```
shreya@shreya-VirtualBox:~$ cat>10.5.txt
#!/bin/bash

if [ -d "mydir" ] ; then
        l=`ls -l mydir|wc -l`
        (( l=l-1 ))
        echo "Total number of files: $l"
else
        mkdir mydir
fi
for i in "$@"
do
        if [ -f "$i" ] ; then
                echo "File $i exists"
        else
                touch "$i"
                mv $i mydir
        fi
done
echo "Contents of mydir: "
echo "$(ls -l mydir)"


shreya@shreya-VirtualBox:~$ cat>hello.txt
hello world
we are in lab
shreya@shreya-VirtualBox:~$ bash 10.5.txt a1.txt hello.txt a2.txt
Total number of files: 3
File hello.txt exists
Contents of mydir:
total 0
-rw-rw-r-- 1 shreya shreya 0 Oct 26 20:52 a1.txt
-rw-rw-r-- 1 shreya shreya 0 Oct 26 20:52 a2.txt
-rw-rw-r-- 1 shreya shreya 0 Oct 26 20:50 hello.txt
```

6. **Get the two file names as arguments and checks whether the two file contents are same or not.**

```
shreya@shreya-VirtualBox:~$ cat>10.6.txt
```

```
#!/bin/bash

f1="$1"
f2="$2"
if diff "$f1" "$f2" > /dev/null ; then
        echo "Both files have same content"
else
        echo "Both files have different content"
fi
```

```
shreya@shreya-VirtualBox:~$ bash 10.6.txt hello.txt hi.txt
Both files have same content
shreya@shreya-VirtualBox:~$ bash 10.6.txt hello.txt 9.8.txt
Both files have different content
shreya@shreya-VirtualBox:~$ bash 10.6.txt hello.txt 10.5.txt
Both files have different content
```

7. To create file and directory if not present.

```
shreya@shreya-VirtualBox:~$ cat>10.7.txt
#!/bin/bash

echo "1.File 2.Directory"
read -p "Enter option: " op
if [ $op -eq 1 ] ; then
        read -p "Enter filename: " filename
        if [ -f $filename ] ; then
                echo "File exists"
        else
                touch $filename
                echo "File created"
        fi
elif [ $op -eq 2 ] ; then
        read -p "Enter directory name: " dirname
        if [ -d $dirname ] ; then
                echo "Directory exists"
        else
                mkdir $dirname
                echo "Directory created"
        fi
else
        echo "Invalid"
fi
```

```
shreya@shreya-VirtualBox:~$ bash 10.7.txt
1.File 2.Directory
Enter option: 1
Enter filename: hello.txt
File exists
shreya@shreya-VirtualBox:~$ bash 10.7.txt
1.File 2.Directory
Enter option: 1
Enter filename: test.txt
File created
shreya@shreya-VirtualBox:~$ bash 10.7.txt
1.File 2.Directory
Enter option: 2
Enter directory name: Desktop
Directory exists

shreya@shreya-VirtualBox:~$ bash 10.7.txt
1.File 2.Directory
Enter option: 2
Enter directory name: unix
Directory created
```

8. To print the number of occurrence of the pattern passed through arguments

```
shreya@shreya-VirtualBox:~$ cat 10.8.txt
#!/bin/bash

read -p "Enter name of file:" fn
read -p "Enter pattern: " pattern
n=`grep -o $pattern $fn | wc -l`
echo "Number of occurences $n"

shreya@shreya-VirtualBox:~$ bash 10.8.txt
Enter name of file:hello.txt
Enter pattern: e
Number of occurences 2

shreya@shreya-VirtualBox:~$ cat hello.txt
hello world
bye
```