1. Write command to show todays date and time in the below given format

i.) DD/MM/YYYY EG: 30/01/2018

```
$ date +"%d/%m/%Y"
                              13/03/2017
```

ii.) DAY, MONTH DATE EG: Wednesday , January 30

```
$ date +"%A,%B %d"
                           Thursday, March 13
```

iii.) DAY DD/MM/YY EG: Wednesday 01/30/13

```
$ date +"%A %d/%m/%y"
                           Thursday 13/03/14
```

iv.) HH:MM:SS EG: 10:30:20

```
$ date +"%T"
                           10:32:13
```

v.) HH:MM AM/PM EG: 10:30 AM

```
$ date +"%R %p"
                           10:38 AM
```

2. Display the calendar of current month

```
$ cal
March 2014
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
```

3. Get the calendar for the month March in the year 2022

```
$ cal 3 2022
March 2022
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
3
```

4. Create three directories letters, reports and assignments under your home directory.

```
$ mkdir letters
$ mkdir reports
$ mkdir assignments
```

a. Move to the directory **letters**.

```
$ cd letters
```

b. Create two directories **friendly** and **formal** under the directory l**etters**.

```
$ mkdir friendly
$ mkdir formal
```

c. Move to directory **reports**. Using single command.

```
$ cd ~/lab/reports
```

d. Create a directory **UNIX** under **assignments** without moving from **reports**.

```
$ mkdir ~/lab/assignments/UNIX
```

e. Move to **UNIX** and check your current directory.

```
$ cd ~/lab/assignments/UNIX
$ pwd
Output: /home/manju/lab/assignments/UNIX
```

f. Create a file **hw4** with a brief report on **man,who,whoami,date.**

```
$ vi hw4
Man command is used to format and display man pages.
Who command is used to determine users who are currently logged in to the Linux operating
system.

Whoami command is used to display the current user name.

Date command is used to display the current date and time.

:wq
```

g. Print the content of **Hw4** from your home directory.

```
$ cat assignments/UNIX/hw4
Output:
Man command is used to format and display man pages.
Who command is used to determine users who are currently logged in to the Linux operating
system.
Whoami command is used to display the current user name.
Date command is used to display the current date and time.
:wq
```

h. Make a copy of **hw4** and store it under the same directory where **hw4** is stored.

```
$ cp assignments/UNIX/hw4  assignments/UNIX/hw5
```

i. Use **ls** command to list the hierarchy in various formats.

```
$ ls -l
Output:
total 12
drwxr-xr-x 3 manju manju 4096 2014-03-13 22:55 assignments
drwxr-xr-x 4 manju manju 4096 2014-03-13 22:54 letters
drwxr-xr-x 2 manju manju 4096 2014-03-13 22:54 reports
```

```
$ ls -a
Output:  …
. assignments letters reports
```

```
$ ls -F
Output:
assignments/ letters/ reports/
```

```
$ ls -R
Output:
.:
assignments letters reports
./assignments:
UNIX
./assignments/UNIX:
hw4 hw5
./letters:
formal friendly
./letters/formal:
./letters/friendly:
./reports:
```

```
$ ls –r
Output:
reports letters assignments
```

```
$ ls -S
Output:
assignments letters reports
```

```
$ ls –A
Output:
assignments letters reports
```

j. Remove **Hw4**.

```
$ rm assignments/UNIX/hw4
```

k. Remove all created folder recursively from your home directory.

```
$ rm -r lab
```

5. Using cat command write a single command to copy file1 to file2

```
cat file1 >> file2
```

6. **Using head command copy lines 1 to 20 of file1 to file2**.

```
head -20 file1 >> file2
```

7. Use the cat command to create the following file. info.txt. (Enter text without Header)

ID	Hr_rate	Hrs_worked
1420	12.56	45
3278	14.56	22
5671	22.54	29
3219	56.7	12
7234	32.44	30
4321	25.09	56
9234	41.22	19

```
cat > info.txt
1420 12.56 45
3278 14.56 22
5671 22.54 29
3219 56.7 12
7234 32.44 30
4321 25.09 56
9234 41.22 19
```

a. Use a command to show number of workers.

```
wc -l info.txt
7 info.txt
```

b. sort the file based on ID & save as S1,txt

```
sort -k1 info.txt
1420 12.56 45
3219 56.7 12
3278 14.56 22
4321 25.09 56
5671 22.54 29
7234 32.44 30
9234 41.22 19
```

c. show the worker who is paid the highest hourly rate.

```
sort -rk2 info.txt | cut -d ' ' -f1 | head -1
3219
```

d. Use a command to show the worker ID who worked more than anybody.

```
sort -rk3 info.txt | cut -d ' ' -f1 | head -1
4321
```

e. Sort the file based on HR_rate and save as S2.txt.

```
sort -k2 info.txt > s2.txt
```

f. Use a command merge the files created on step b and e. call it as S.txt

```
cat s1.txt s2.txt > s.txt
```

g. Reverse the file line by line. Last line becomes first line and second last line becomes second and so on. Call it as info_rev.txt

```
tac info.txt > info_rev.txt
```

8. Count lines, words, and characters in a file with the **wc** command.

```
wc -lwc file1
11 33 229 file1
```

9. Write *grep* commands to do the following activities: Create a file fileg1.txt

a. To select the lines from a file that has exactly two characters.

```
grep "^..$" sample
if
```

b. To select the lines from a file that start with the upper case letter.

```
grep "[A-Z]*" sample
How
Fine
Shibin
```

c. To select lines from a file that has at least three characters.

```
grep "^..." sample
hai naveen
Fine
Think
```

d. To select lines from a file that has at least two digits without any other characters in between.

```
grep "[0-9][0-9]" sample
u 23
2er334 thnk
```

11. Shell script to compare two strings entered from the keyboard

```
#!/bin/bash
echo "Enter String 1"
read nam1
echo "Enter String 2"
read nam2
if [ "$nam1" == "$nam2" ];
then
echo "Equal"
else
echo "NOT Equal"
fi
```

OUTPUT

```
bash 1
Enter String 1
manu
Enter String 2
manu
Equal
```

12. Shell script which would display the summation of the digits of the given number

```
#!/bin/bash
echo "Enter a number"
read num
sd=0
sum=0
while [ $num -gt 0 ]
do
sd=$(($num % 10))
num=$(($num / 10))
sum=$(($sum + $sd))
```

```
done
echo "sum of all digit is $sum"
```

OUTPUT

```
bash 3
Enter a number
45
sum of all digit is 9
```

13. Shell script to reverse a given number

```
#!/bin/bash
echo "Enter a number"
read num
sd=0
rev=0
while [ $num -gt 0 ]
do
sd=$(($num % 10))
rev=$(($rev * 10 + $sd))
num=$(($num / 10))
done
echo "Reverse is $rev"
```

OUTPUT

```
bash 4
Enter a number
234
Reverse is 432
```

14. Shell script to simulate a simple calculator using integer arithmetic operations

```
#!/bin/bash
sum=0;
i="y"
echo "Enter one Number"
read n1
echo "Enter Second Number"
read n2
while [ 1 ]
do
echo "1. Addition"
echo "2. Substraction"
echo "3. Multiplication"
echo "4. Division"
echo "Enter your Choice"
read ch
case $ch in
1) sum=` expr $n1 + $n2 `
echo "Sum="$sum;;
2) sum=` expr $n1 - $n2 `
echo "Sub="$sum;;
3) sum=` expr $n1 \* $n2 `
echo "Multi="$sum;;
4) sum=` expr $n1 / $n2 `
echo "div="$sum;;
*) echo "invalid option";;
esac
echo "To exit press 0 else press 1"
read i
if [ $i -eq 0 ]
then
exit
fi
done
```

OUTPUT

```
bash 5
Enter one Number
4
Enter Second Number
2
1. Addition
2. Substraction
3. Multiplication
4. Division
Enter your Choice
```

```
4
div=2
To exit press 0 else press 1
0
```