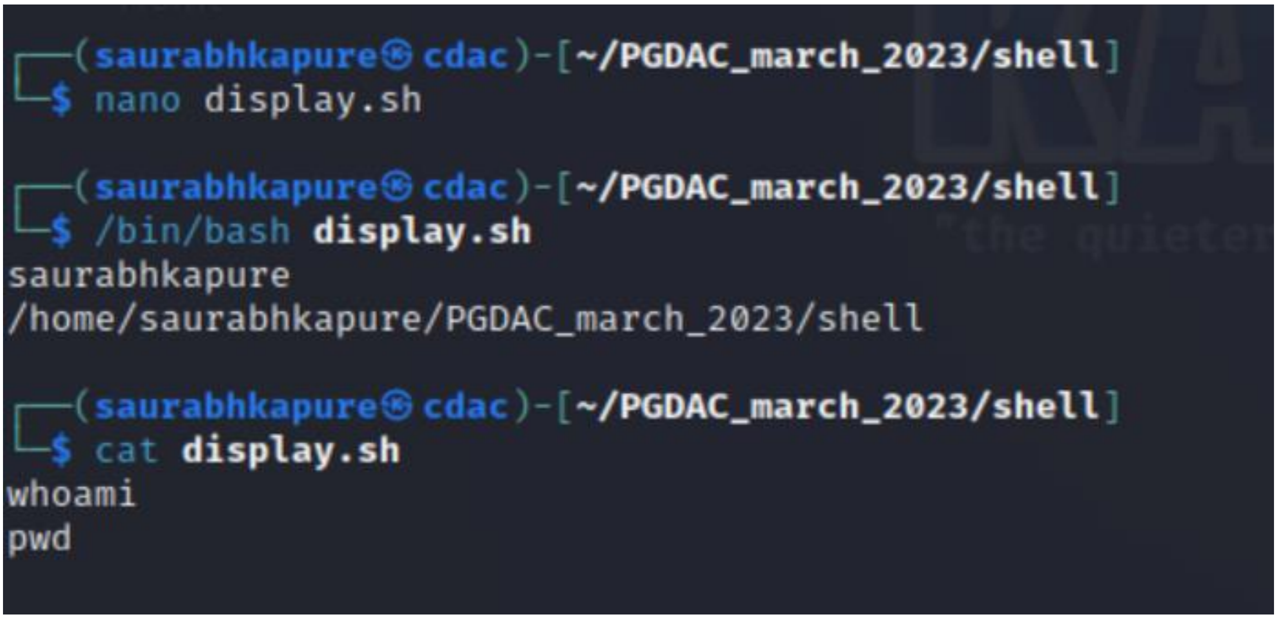
Name = SAURABH SHARAD KAPURE

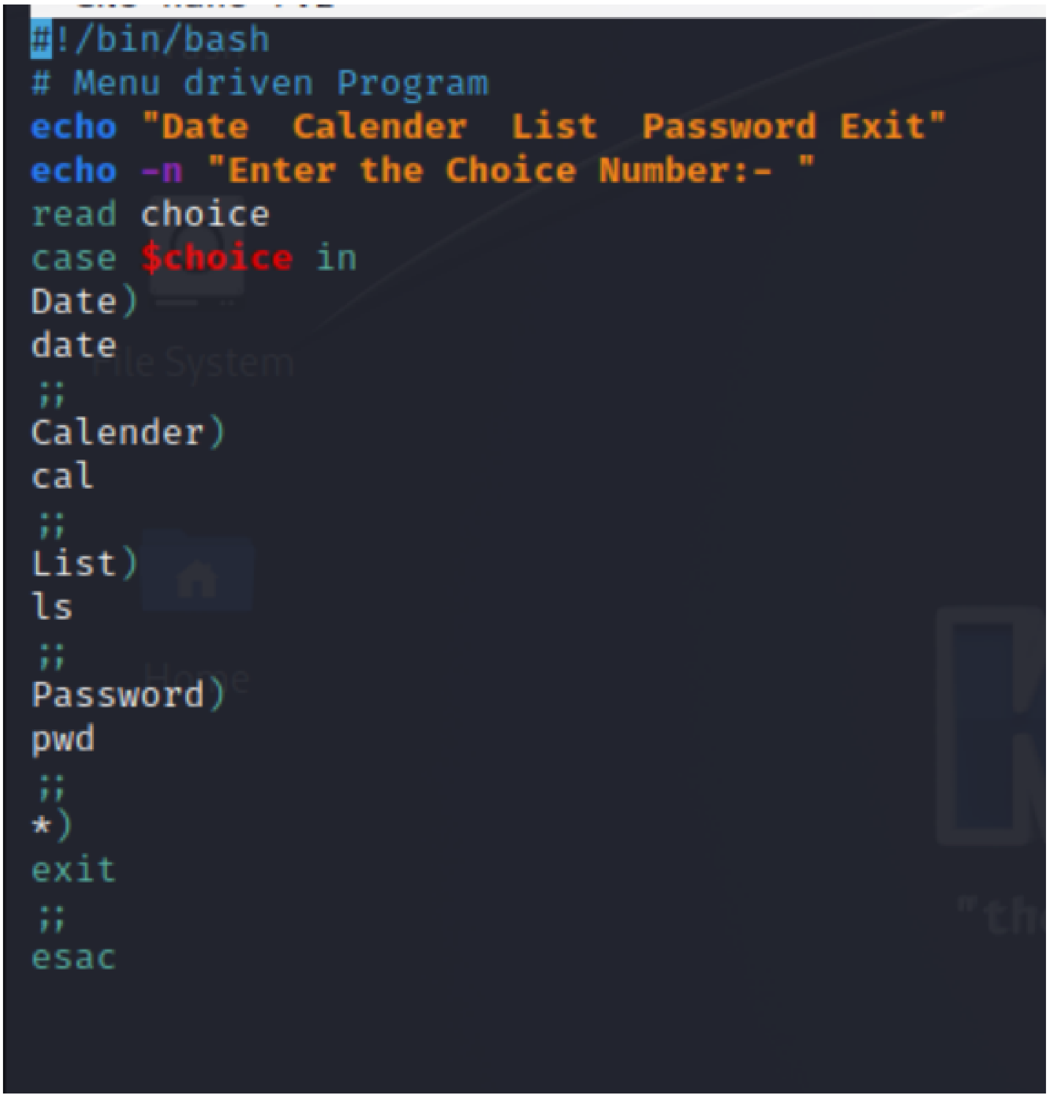
Shell Assignment No = 1

Q1. Write a shell script to display your LOGIN NAME and HOME directory.



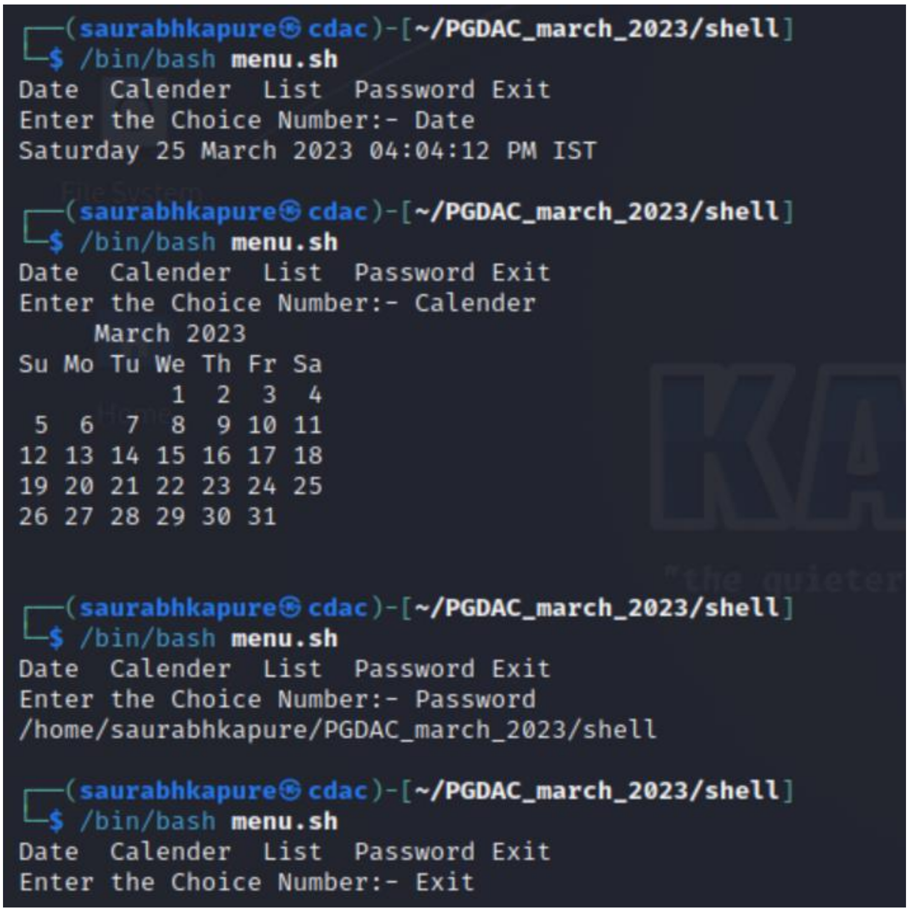
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano display.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash display.sh
saurabhkapure
/home/saurabhkapure/PGDAC_march_2023/shell

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ cat display.sh
whoami
pwd
```

Q2. Write a shell script to display menu like “1. Date, 2. Cal, 3. Ls, 4. Pwd, 5. Exit” and execute the commands depending on user choice.



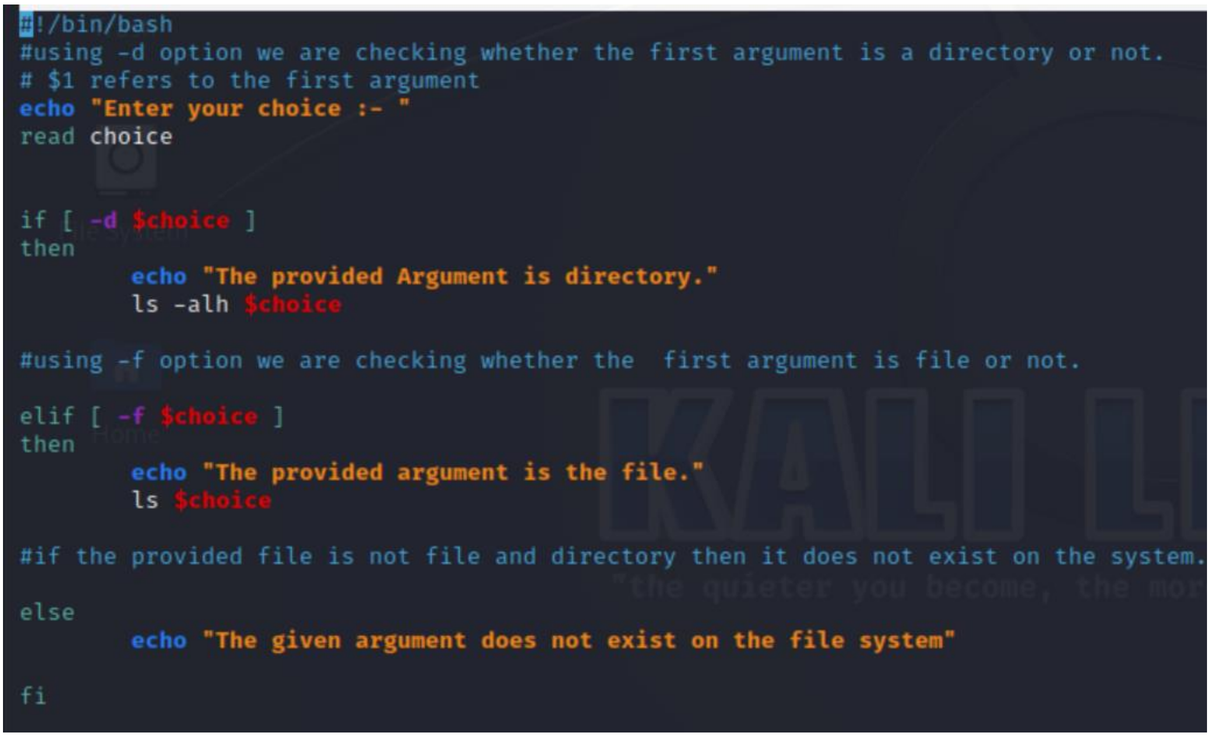
```
#!/bin/bash
# Menu driven Program
echo "Date  Calender  List  Password Exit"
echo -n "Enter the Choice Number:- "
read choice
case $choice in
Date)
date
;;
Calender)
cal
;;
List)
ls
;;
Password)
pwd
;;
*)
exit
;;
esac
```

Output :-



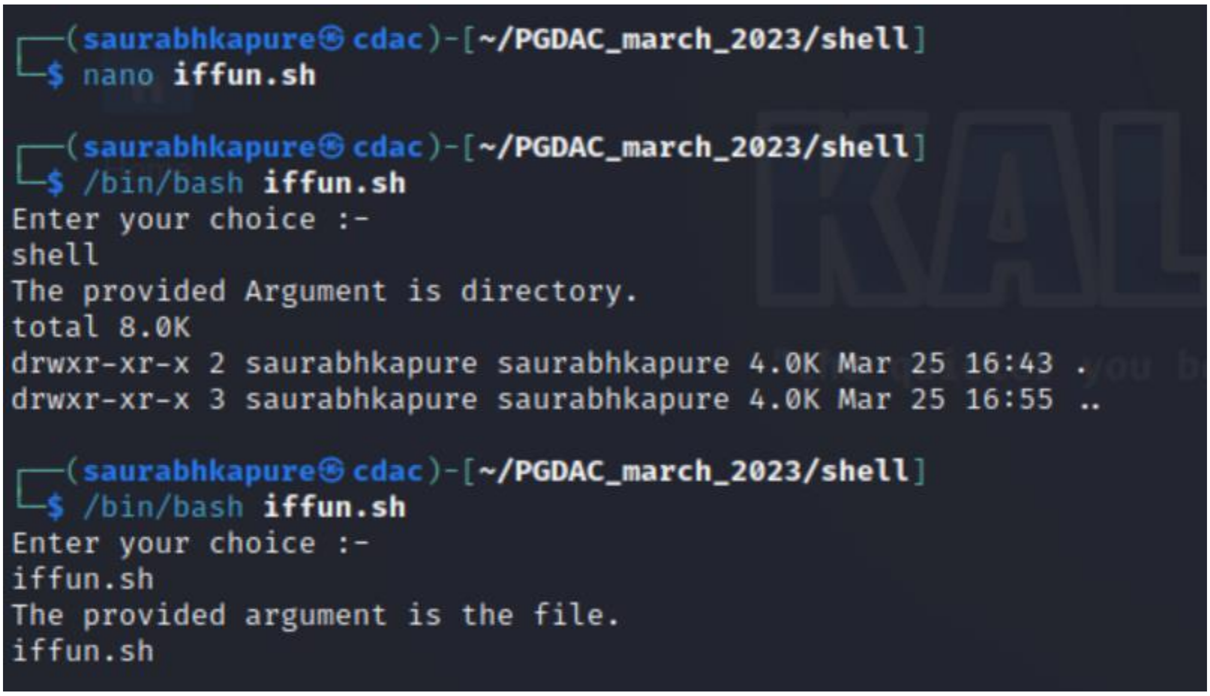
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash menu.sh
Date  Calender  List  Password Exit
Enter the Choice Number:- Date
Saturday 25 March 2023 04:04:12 PM IST

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash menu.sh
Date  Calender  List  Password Exit
Enter the Choice Number:- Calender
     March 2023
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


┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash menu.sh
Date  Calender  List  Password Exit
Enter the Choice Number:- Password
/home/saurabhkapure/PGDAC_march_2023/shell

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash menu.sh
Date  Calender  List  Password Exit
Enter the Choice Number:- Exit
```

Q3. Write a shell script to accept the name from the user and check whether user entered name



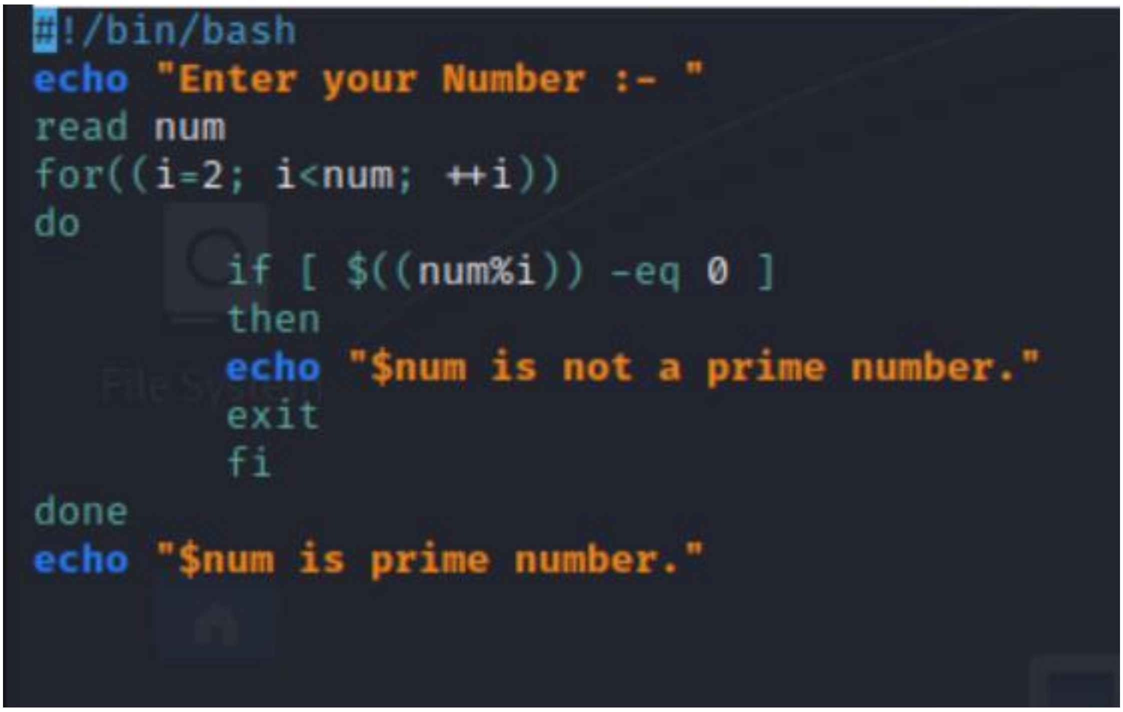
```
#!/bin/bash
#using -d option we are checking whether the first argument is a directory or not.
# $1 refers to the first argument
echo "Enter your choice :- "
read choice


if [ -d $choice ]
then
        echo "The provided Argument is directory."
        ls -alh $choice

#using -f option we are checking whether the  first argument is file or not.

elif [ -f $choice ]
then
        echo "The provided argument is the file."
        ls $choice

#if the provided file is not file and directory then it does not exist on the system.

else
        echo "The given argument does not exist on the file system"

fi
```

Output :-



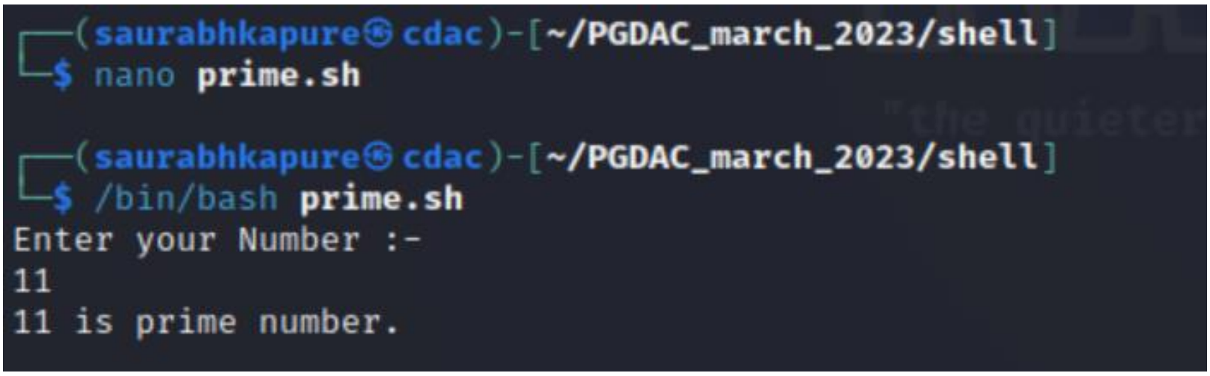
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano iffun.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash iffun.sh
Enter your choice :-
shell
The provided Argument is directory.
total 8.0K
drwxr-xr-x 2 saurabhkapure saurabhkapure 4.0K Mar 25 16:43 .
drwxr-xr-x 3 saurabhkapure saurabhkapure 4.0K Mar 25 16:55 ..

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash iffun.sh
Enter your choice :-
iffun.sh
The provided argument is the file.
iffun.sh
```

Q4. Write a shell script to determine whether a given number is prime or not.?



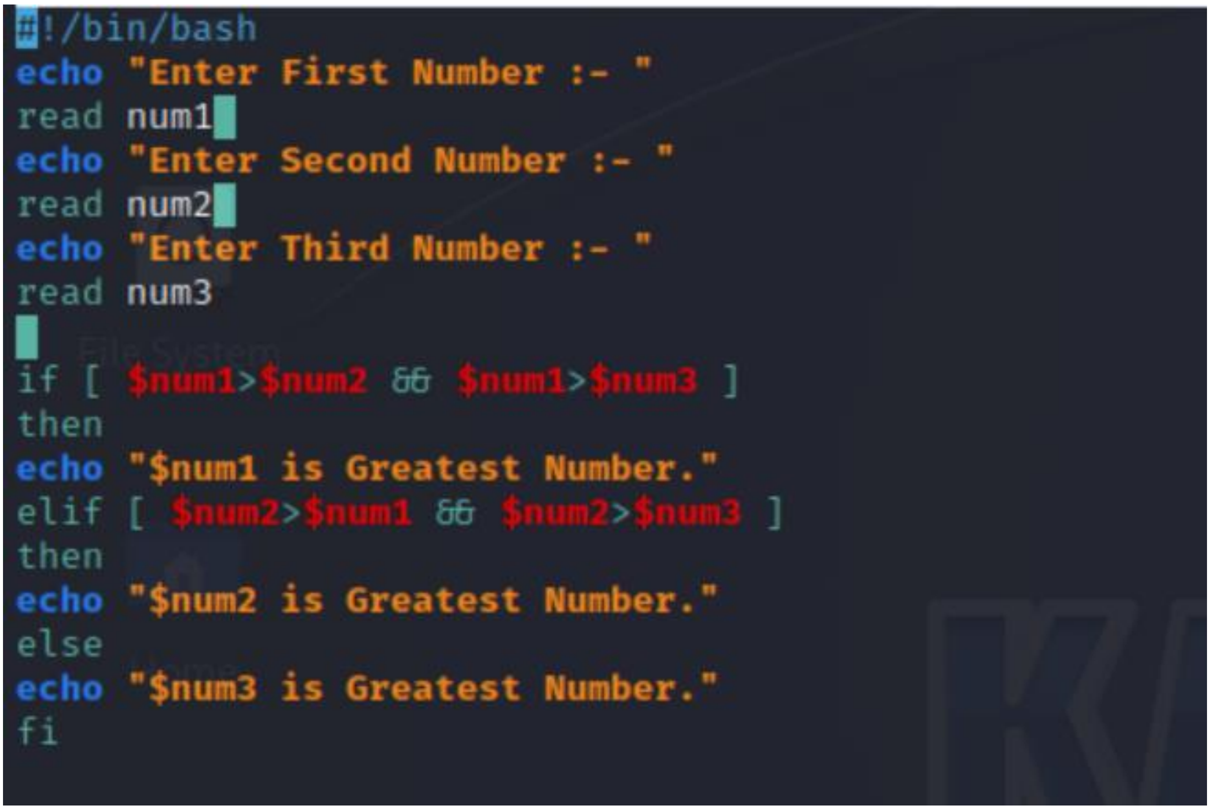
```
#!/bin/bash
echo "Enter your Number :- "
read num
for((i=2; i<num; ++i))
do
        if [ $((num%i)) -eq 0 ]
        then
        echo "$num is not a prime number."
        exit
        fi
done
echo "$num is prime number."
```

Output :-



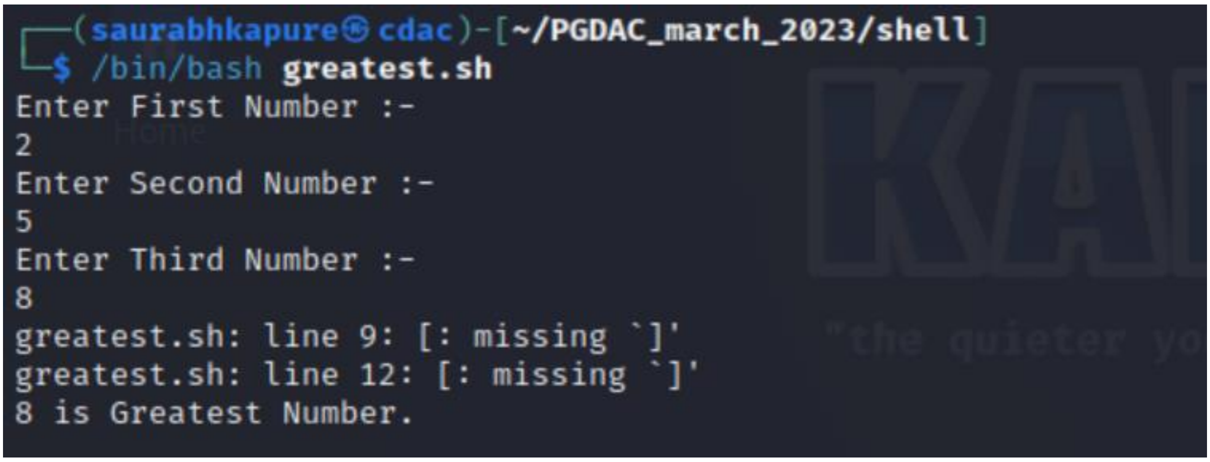
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano prime.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash prime.sh
Enter your Number :-
11
11 is prime number.
```

Q5. Write a program to find the greatest of three numbers.?



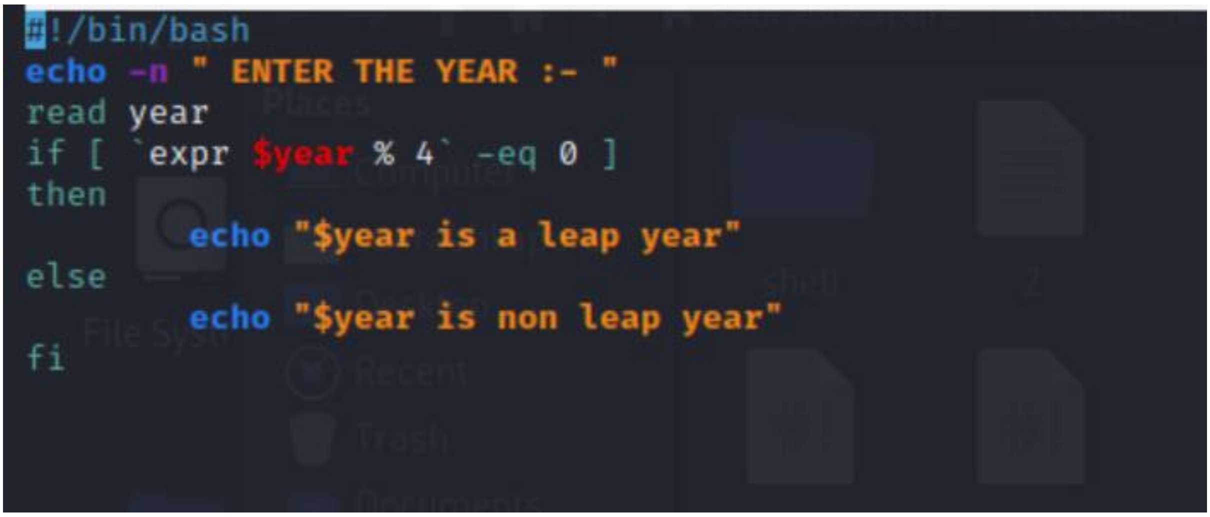
```bash
#!/bin/bash
echo "Enter First Number :- "
read num1
echo "Enter Second Number :- "
read num2
echo "Enter Third Number :- "
read num3

if [ $num1>$num2 && $num1>$num3 ]
then
echo "$num1 is Greatest Number."
elif [ $num2>$num1 && $num2>$num3 ]
then
echo "$num2 is Greatest Number."
else
echo "$num3 is Greatest Number."
fi
```

Output :-



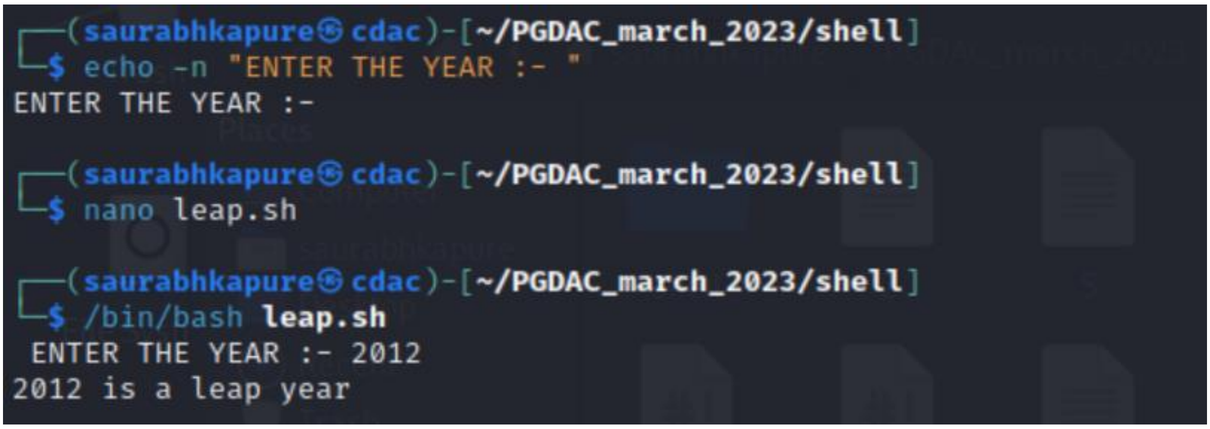
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash greatest.sh
Enter First Number :-
2
Enter Second Number :-
5
Enter Third Number :-
8
greatest.sh: line 9: [: missing `]'
greatest.sh: line 12: [: missing `]'
8 is Greatest Number.
```

Q6. Write a program to find whether a given year is a leap year or not.?



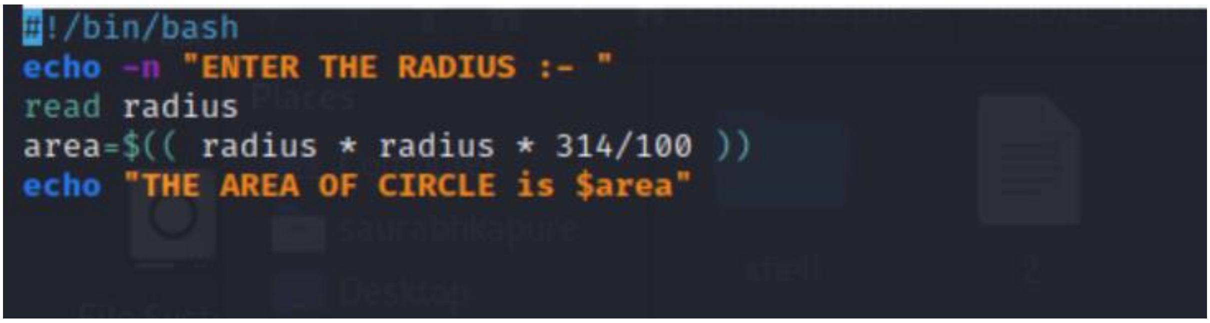
```
#!/bin/bash
echo -n " ENTER THE YEAR :- "
read year
if [ `expr $year % 4` -eq 0 ]
then
        echo "$year is a leap year"
else
        echo "$year is non leap year"
fi
```

Output :-



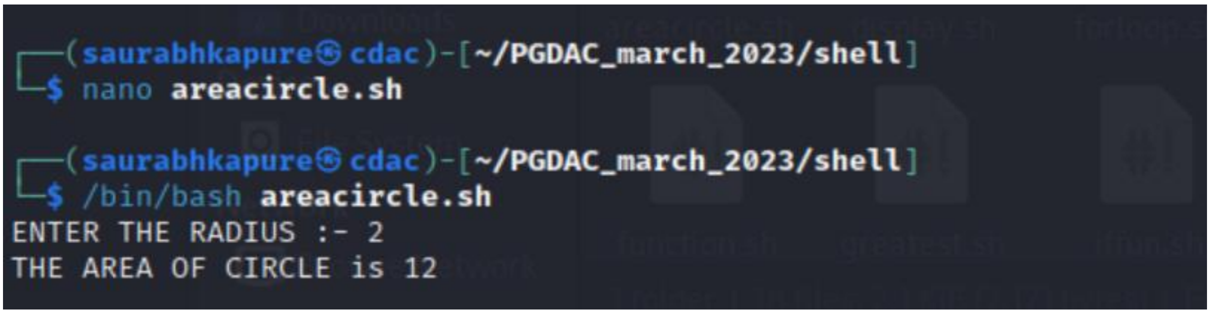
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ echo -n "ENTER THE YEAR :- "
ENTER THE YEAR :-

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano leap.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash leap.sh
 ENTER THE YEAR :- 2012
2012 is a leap year
```

Q7. Write a program to find out the area of a circle.?



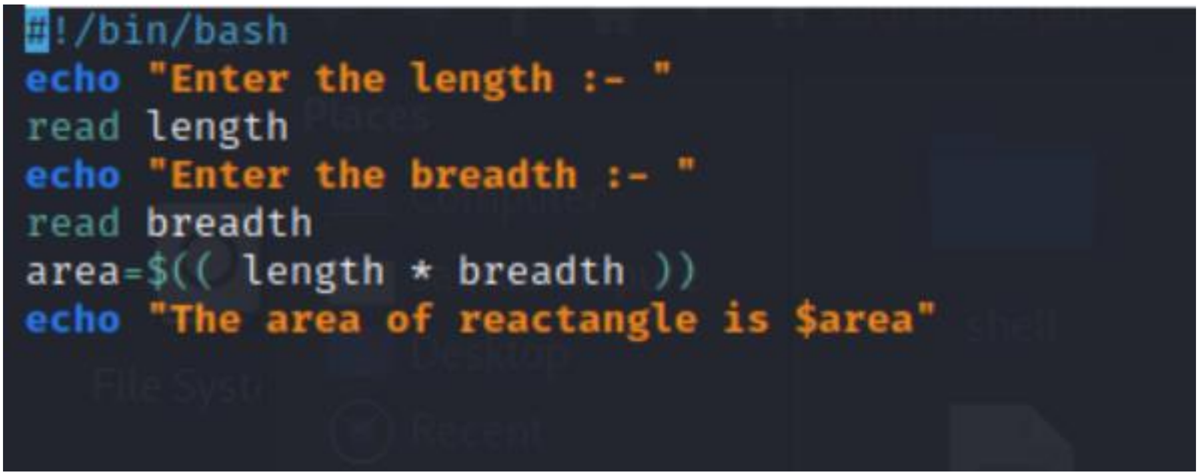
```
#!/bin/bash
echo -n "ENTER THE RADIUS :- "
read radius
area=$(( radius * radius * 314/100 ))
echo "THE AREA OF CIRCLE is $area"
```

Output :-



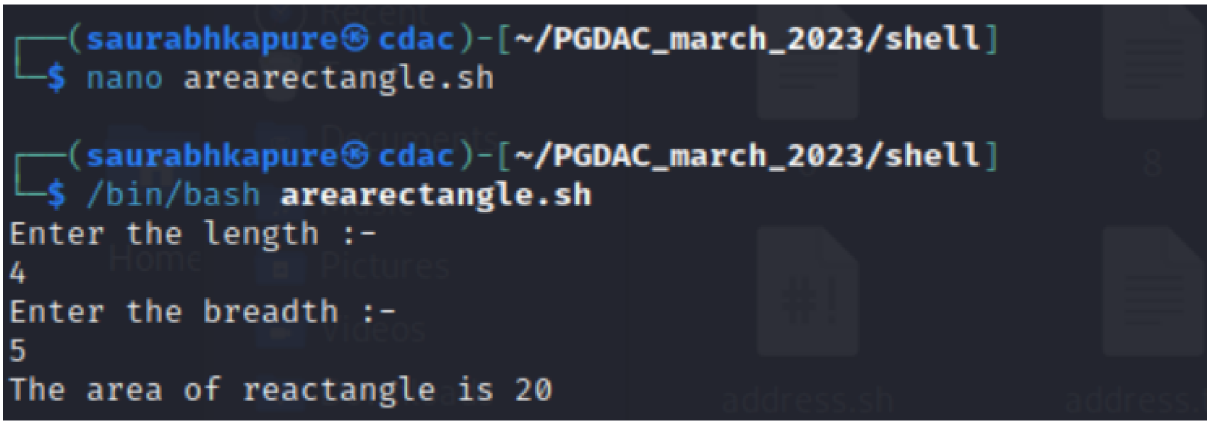
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano areacircle.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash areacircle.sh
ENTER THE RADIUS :- 2
THE AREA OF CIRCLE is 12
```

Q8. Write a program to find out the area of a rectangle.?



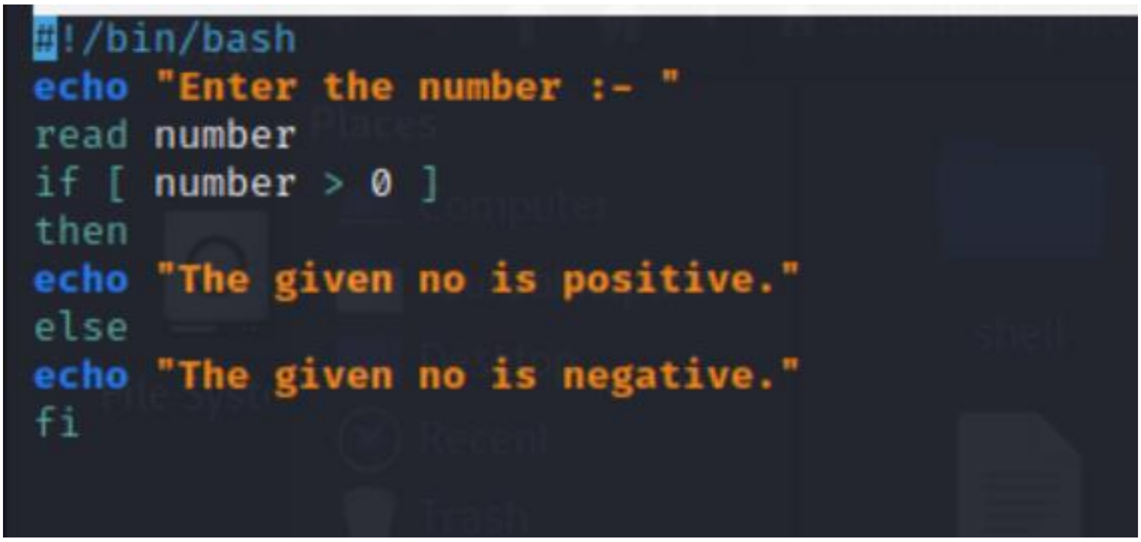
```
#!/bin/bash
echo "Enter the length :- "
read length
echo "Enter the breadth :- "
read breadth
area=$(( length * breadth ))
echo "The area of reactangle is $area"
```

Output :-



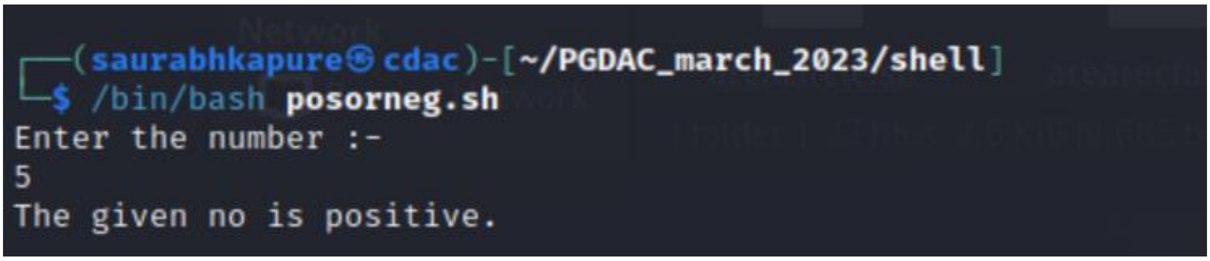
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano arearectangle.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash arearectangle.sh
Enter the length :-
4
Enter the breadth :-
5
The area of reactangle is 20
```

Q9. Write a program to find whether a given number is positive or negative



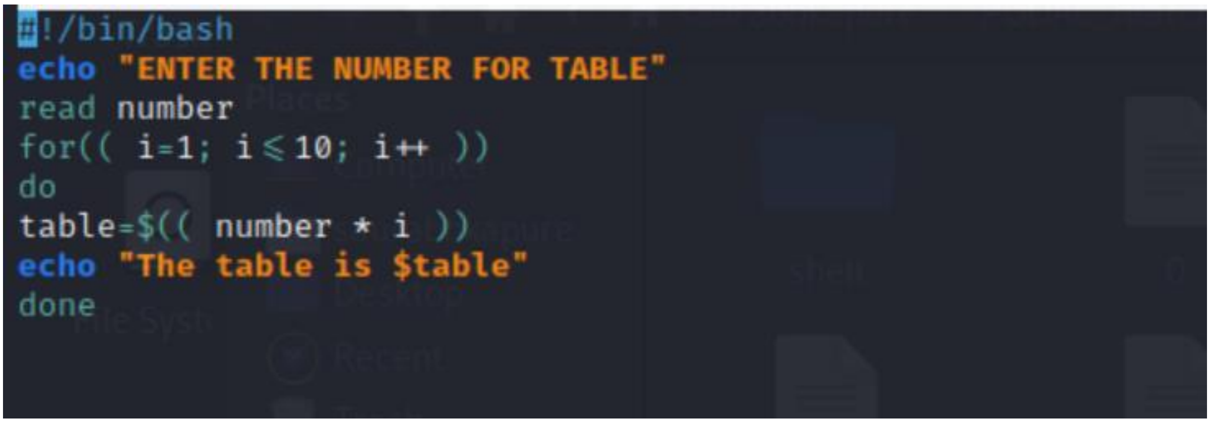
```
#!/bin/bash
echo "Enter the number :- "
read number
if [ number > 0 ]
then
echo "The given no is positive."
else
echo "The given no is negative."
fi
```

Output :-



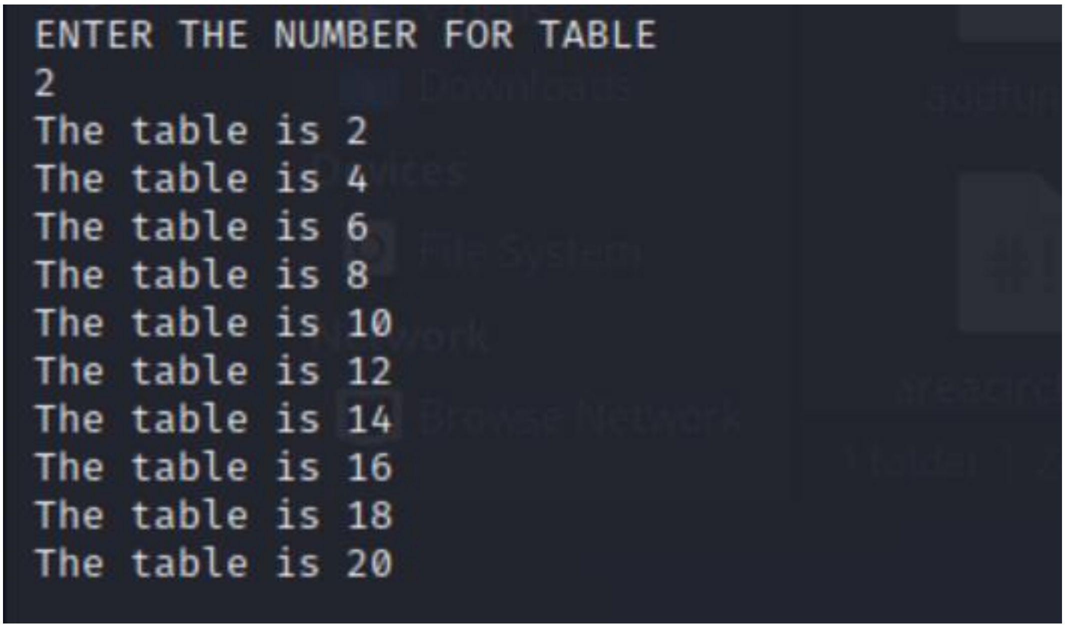
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash posorneg.sh
Enter the number :-
5
The given no is positive.
```

Q.10 Write a program to print the table of a given number.?



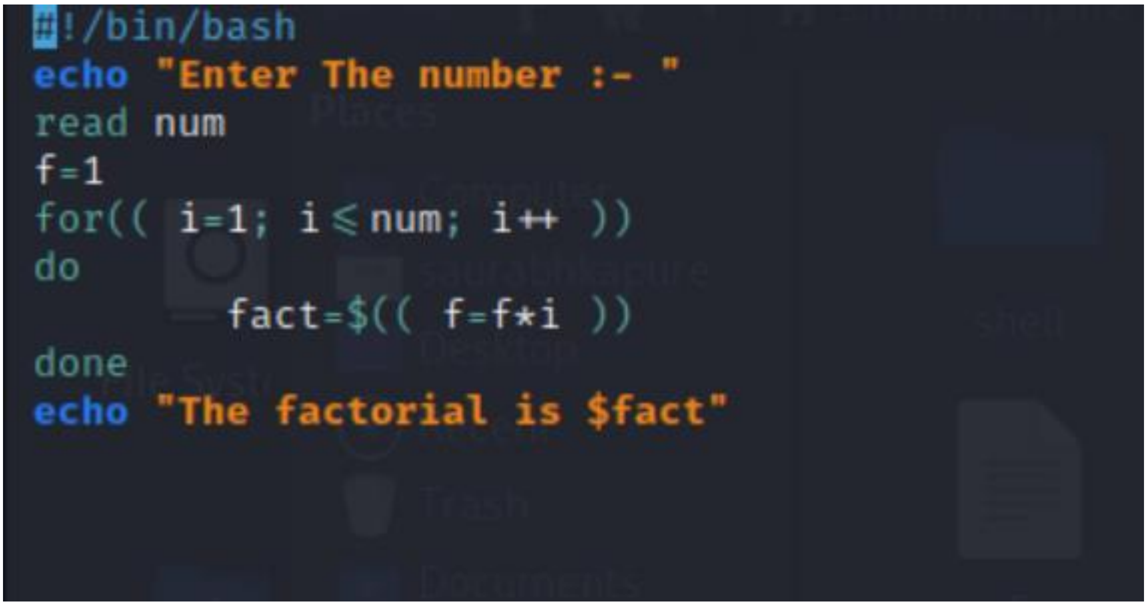
```
#!/bin/bash
echo "ENTER THE NUMBER FOR TABLE"
read number
for(( i=1; i<=10; i++ ))
do
table=$(( number * i ))
echo "The table is $table"
done
```

Output :-



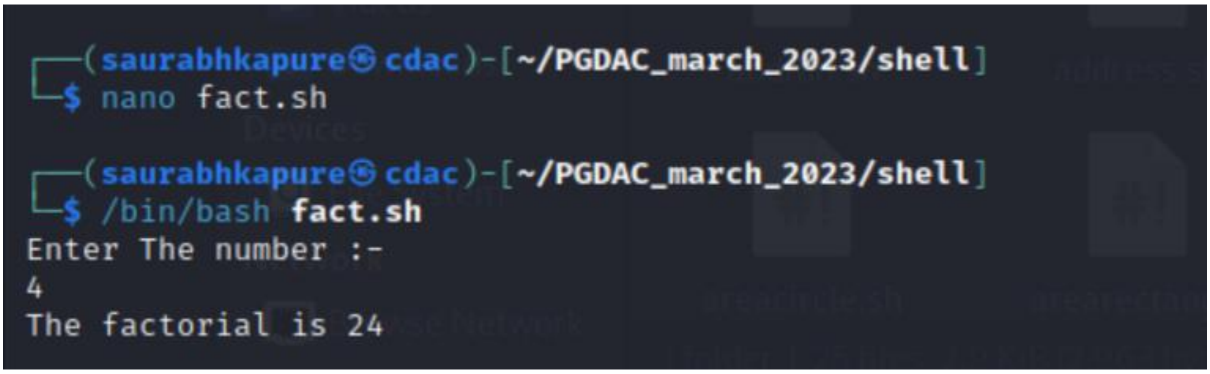
```
ENTER THE NUMBER FOR TABLE
2
The table is 2
The table is 4
The table is 6
The table is 8
The table is 10
The table is 12
The table is 14
The table is 16
The table is 18
The table is 20
```

Q.11 Write a program to find the factorial of given number.



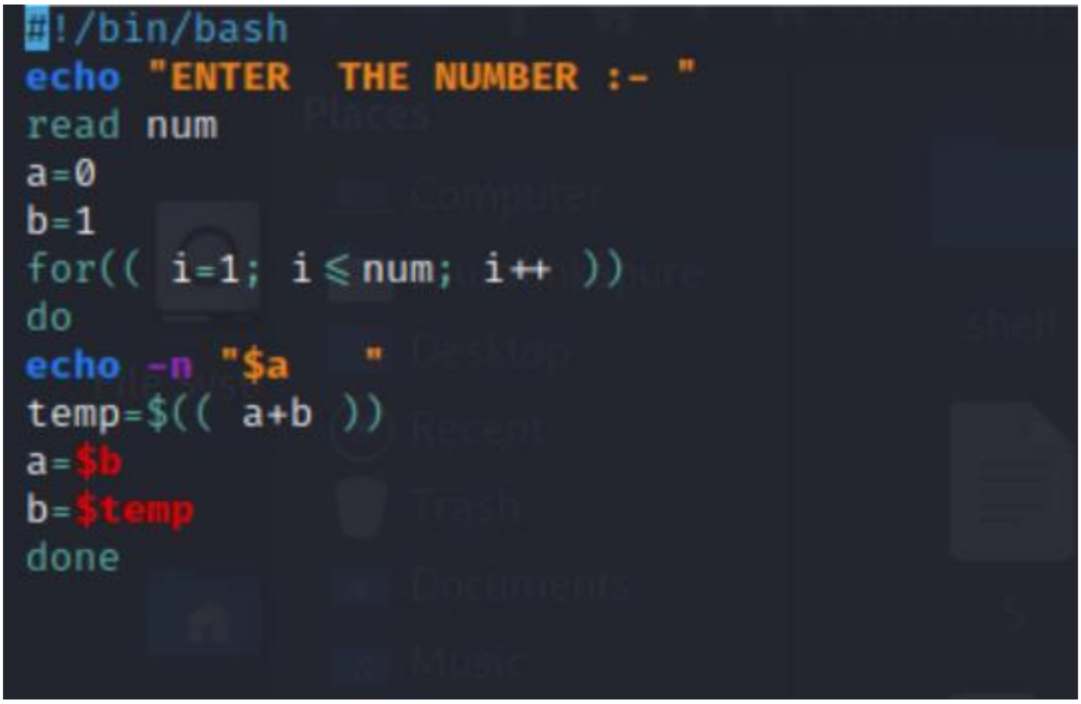
```bash
#!/bin/bash
echo "Enter The number :- "
read num
f=1
for(( i=1; i<=num; i++ ))
do
        fact=$(( f=f*i ))
done
echo "The factorial is $fact"
```

Output :-



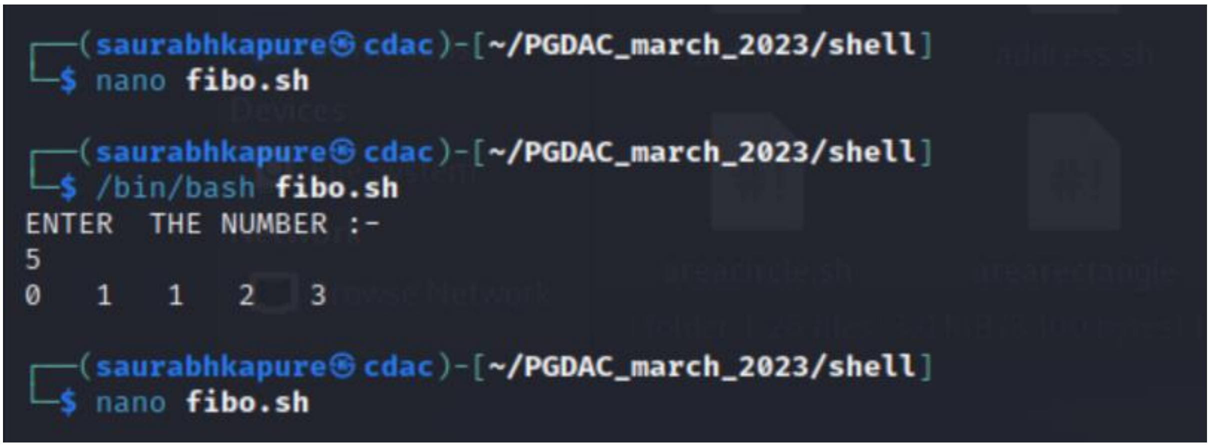
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano fact.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash fact.sh
Enter The number :-
4
The factorial is 24
```

Q12. Write a program to find given number of terms in the Fibonacci series.



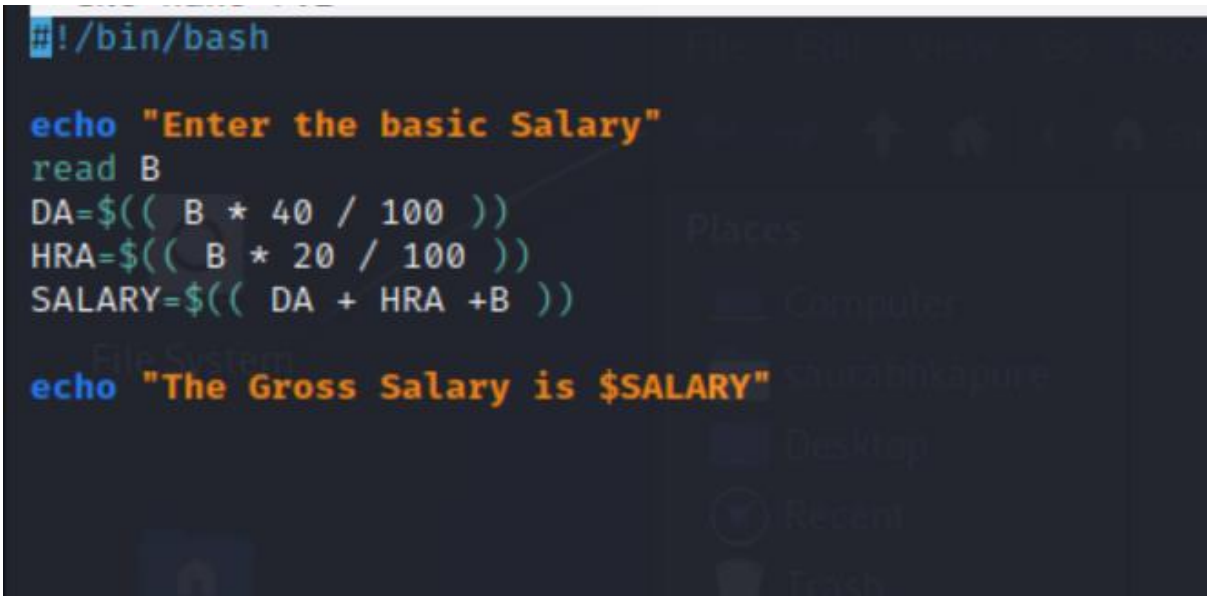
```
#!/bin/bash
echo "ENTER  THE NUMBER :- "
read num
a=0
b=1
for(( i=1; i<=num; i++ ))
do
echo -n "$a    "
temp=$(( a+b ))
a=$b
b=$temp
done
```

Output :-



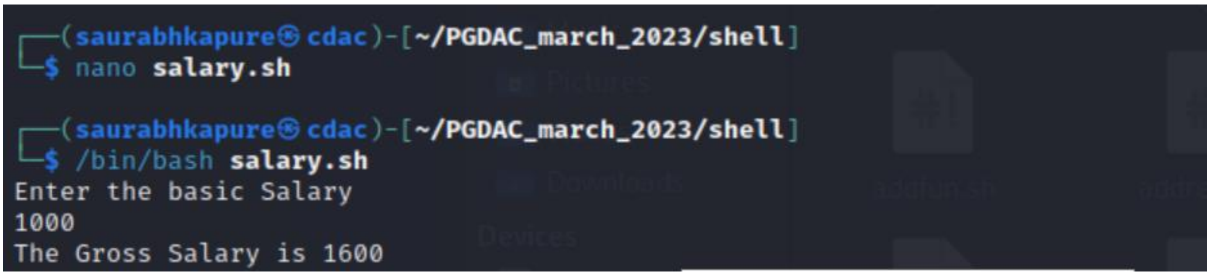
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano fibo.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash fibo.sh
ENTER  THE NUMBER :-
5
0    1    1    2    3

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano fibo.sh
```

Q13. Write a program to calculate gross salary if the DA is 40%, HRA is 20% of basic salary. Accept basic salary form user and display gross salary (Result can be floating point value).



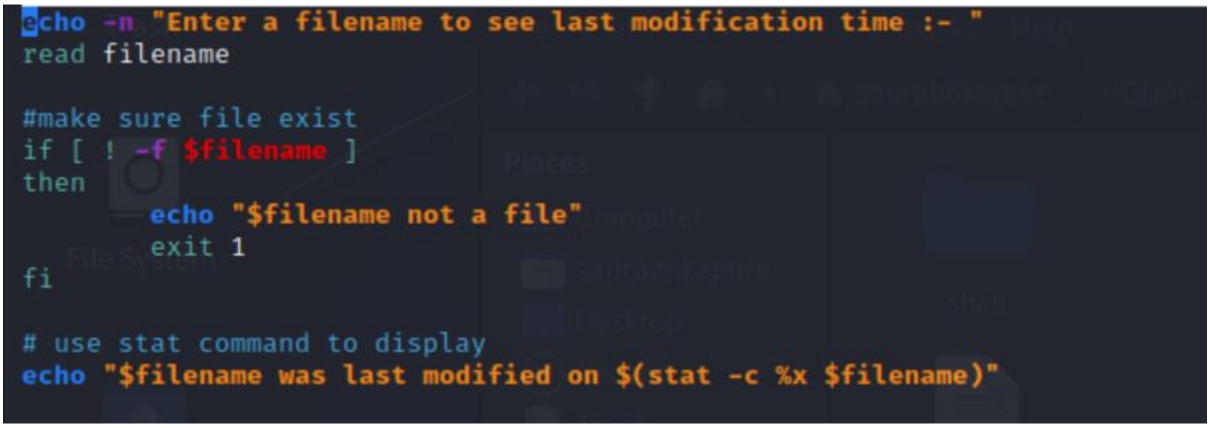
```
#!/bin/bash

echo "Enter the basic Salary"
read B
DA=$(( B * 40 / 100 ))
HRA=$(( B * 20 / 100 ))
SALARY=$(( DA + HRA +B ))

echo "The Gross Salary is $SALARY"
```

Output :-



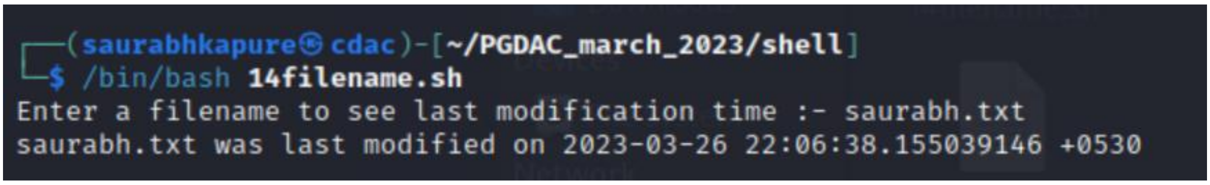
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano salary.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash salary.sh
Enter the basic Salary
1000
The Gross Salary is 1600
```

Q14. Write a shell script to accept a filename as argument and displays the last modification time if the file exists and a suitable message if it doesn’t exist.



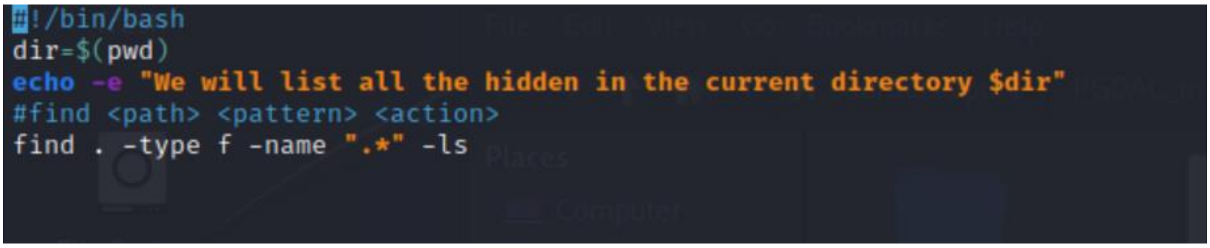
```
echo -n "Enter a filename to see last modification time :- "
read filename

#make sure file exist
if [ ! -f $filename ]
then
        echo "$filename not a file"
        exit 1
fi

# use stat command to display
echo "$filename was last modified on $(stat -c %x $filename)"
```

Output :-



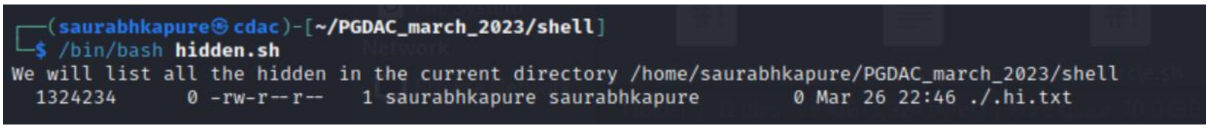
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash 14filename.sh
Enter a filename to see last modification time :- saurabh.txt
saurabh.txt was last modified on 2023-03-26 22:06:38.155039146 +0530
```

15. Write a shell script to display only hidden file of current directory.



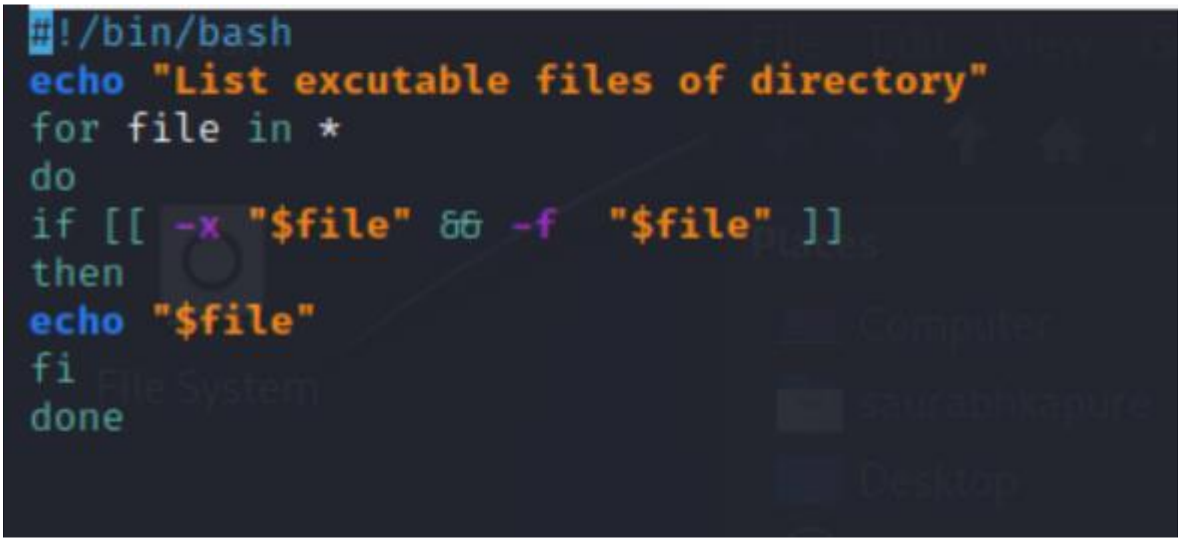
```
#!/bin/bash
dir=$(pwd)
echo -e "We will list all the hidden in the current directory $dir"
#find <path> <pattern> <action>
find . -type f -name ".*" -ls
```

Output :-



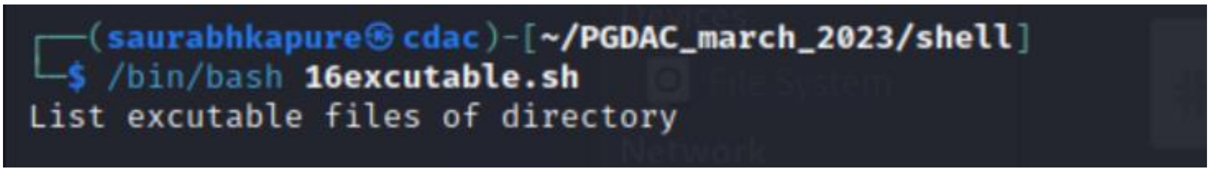
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash hidden.sh
We will list all the hidden in the current directory /home/saurabhkapure/PGDAC_march_2023/shell
  1324234      0 -rw-r--r--   1 saurabhkapure saurabhkapure        0 Mar 26 22:46 ./.hi.txt
```

Q16. Write a shell script to display only executable files of current directory.



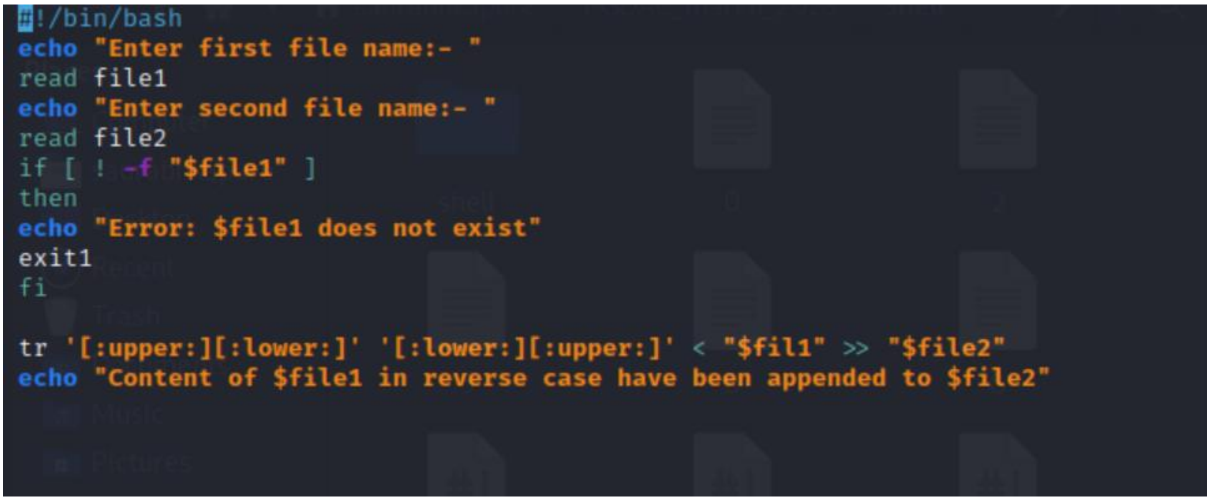
```
#!/bin/bash
echo "List excutable files of directory"
for file in *
do
if [[ -x "$file" && -f  "$file" ]]
then
echo "$file"
fi
done
```

Output :-



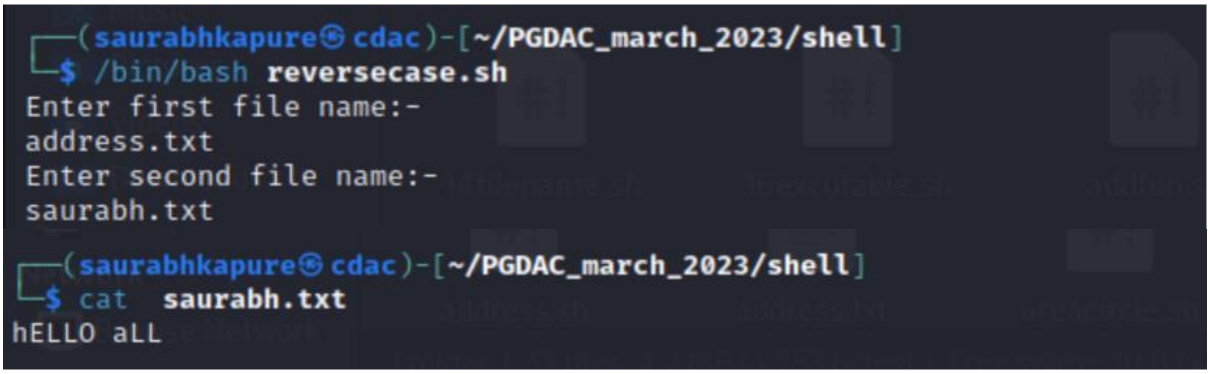
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash 16excutable.sh
List excutable files of directory
```

17. Accept the two file names from user and append the contents in reverse case of first file into second file.



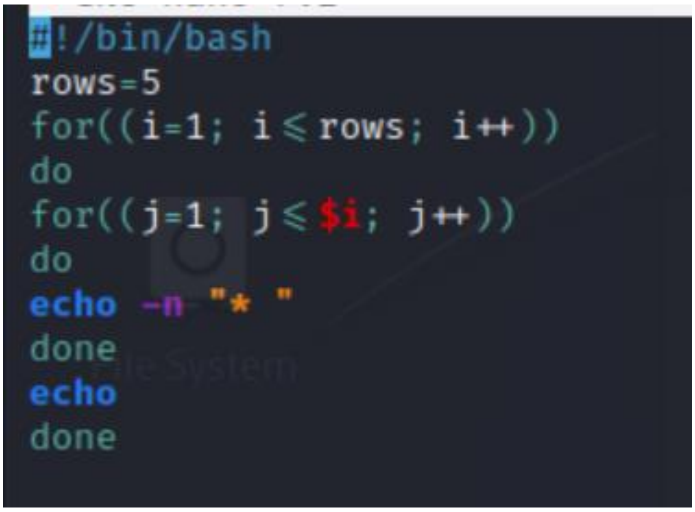
```
#!/bin/bash
echo "Enter first file name:- "
read file1
echo "Enter second file name:- "
read file2
if [ ! -f "$file1" ]
then
echo "Error: $file1 does not exist"
exit1
fi

tr '[:upper:][:lower:]' '[:lower:][:upper:]' < "$fil1" >> "$file2"
echo "Content of $file1 in reverse case have been appended to $file2"
```

Output :-



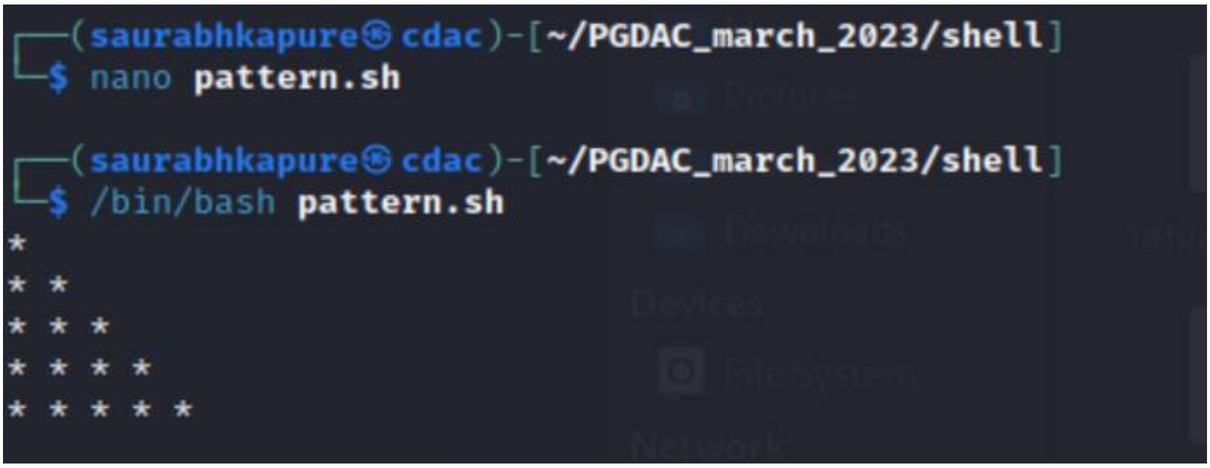
```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash reversecase.sh
Enter first file name:-
address.txt
Enter second file name:-
saurabh.txt

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ cat  saurabh.txt
hELLO aLL
```

Q18. Print the following pattern.



```
#!/bin/bash
rows=5
for((i=1; i<=rows; i++))
do
for((j=1; j<=$i; j++))
do
echo -n "* "
done
echo
done
```

Output :-



```
┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ nano pattern.sh

┌──(saurabhkapure㉿cdac)-[~/PGDAC_march_2023/shell]
└─$ /bin/bash pattern.sh
*
* *
* * *
* * * *
* * * * *
```