DATE:

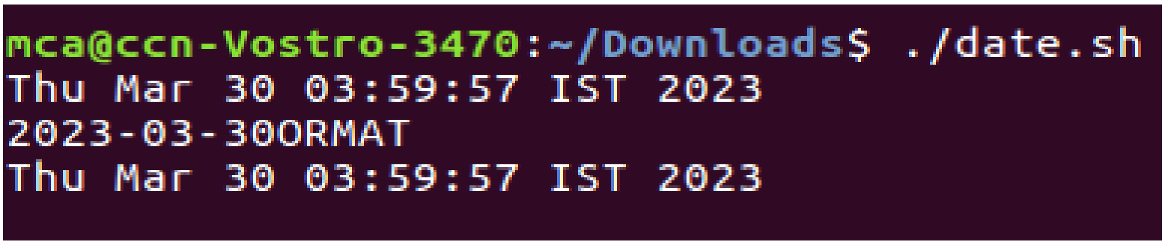
BASIC SHELL SCRIPT PROGRAM QUESTIONS

1. Write a script to show current date, time and current directory.

SOURCE CODE

```
#!/bin/bash
date
date +"%FORMAT"
var=$(date)
var=`date`
echo "$var"
echo $pwd
```

OUTPUT



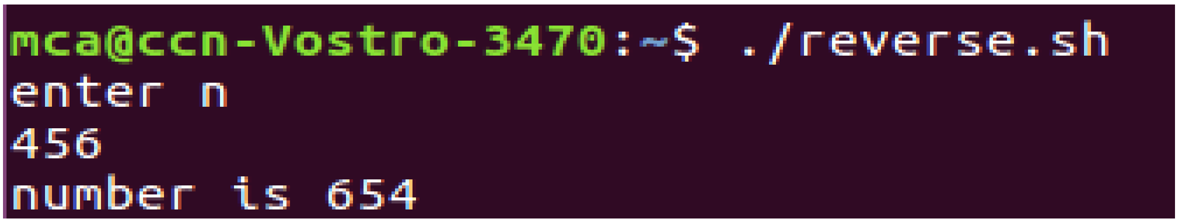
```
mca@ccn-Vostro-3470:~/Downloads$ ./date.sh
Thu Mar 30 03:59:57 IST 2023
2023-03-30ORMAT
Thu Mar 30 03:59:57 IST 2023
```

2. Write a script to reverse of a number

SOURCE CODE

```
#!/bin/bash
echo enter n
read n
num=0
while [ $n -gt 0 ]
do
num=$(expr $num \* 10)
k=$(expr $n % 10)
num=$(expr $num + $k)
n=$(expr $n / 10)
done
echo number is $num
```

OUTPUT



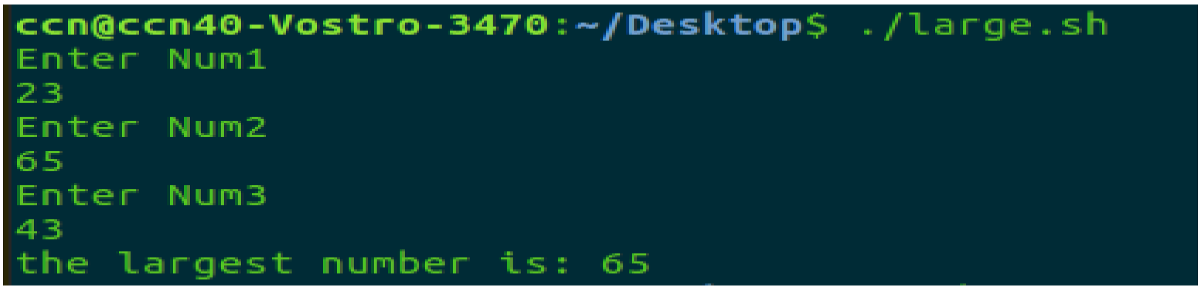
```
mca@ccn-Vostro-3470:~$ ./reverse.sh
enter n
456
number is 654
```

3.Write a script to largest among three numbers

SOURCE CODE

```
#!/bin/bash
echo "Enter Num1"
read num1
echo "Enter Num2"
read num2
echo "Enter Num3"
read num3
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
   echo $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
   echo $num2
else
   echo $num3
fi
```

OUTPUT



```
ccn@ccn40-Vostro-3470:~/Desktop$ ./large.sh
Enter Num1
23
Enter Num2
65
Enter Num3
43
the largest number is: 65
```

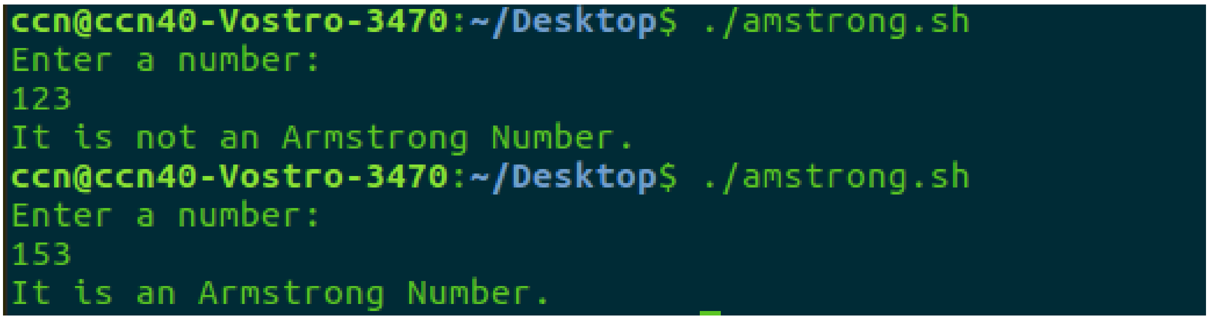
3. Write a script check whether the number is Armstrong or not.

SOURCE CODE

```
#!/bin/bash
echo "Enter a number: "
read c
x=$c
sum=0
r=0
n=0
while [ $x -gt 0 ]
do
r=`expr $x % 10`
n=`expr $r \* $r \* $r`
sum=`expr $sum + $n`
x=`expr $x / 10`
done
if [ $sum -eq $c ]
then
echo "It is an Armstrong Number."
```

```
 else
 echo "It is not an Armstrong Number."
 fi
```

OUTPUT



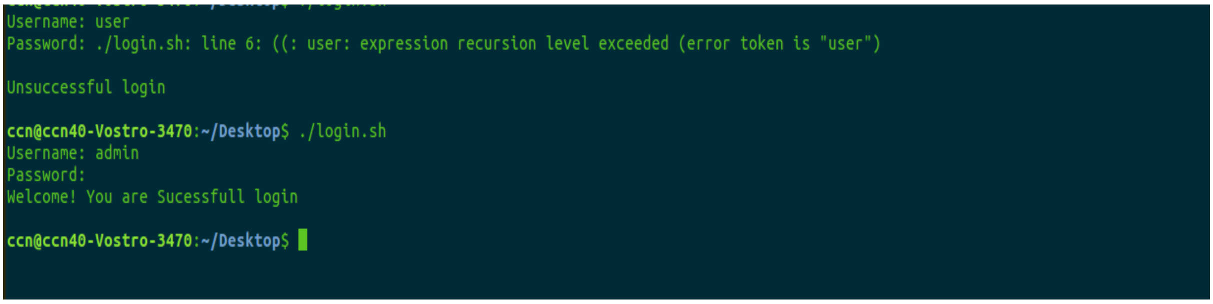
```
ccn@ccn40-Vostro-3470:~/Desktop$ ./amstrong.sh
Enter a number:
123
It is not an Armstrong Number.
ccn@ccn40-Vostro-3470:~/Desktop$ ./amstrong.sh
Enter a number:
153
It is an Armstrong Number.
```

5. Write a script to check password and login

SOURCE CODE

```
# !/bin/bash
#Type your login details
read -p 'Username: ' user
read -sp 'Password: ' pass
if (( $user == "admin" && $pass == "admin123" ))
then
  echo -e "\nWelcome! You are Sucessfull login\n"
 else
  echo -e "\nUnsuccessful login\n"
fi
```

OUTPUT



```
Username: user
Password: ./login.sh: line 6: ((: user: expression recursion level exceeded (error token is "user")

Unsuccessful login

ccn@ccn40-Vostro-3470:~/Desktop$ ./login.sh
Username: admin
Password:
Welcome! You are Sucessfull login

ccn@ccn40-Vostro-3470:~/Desktop$
```

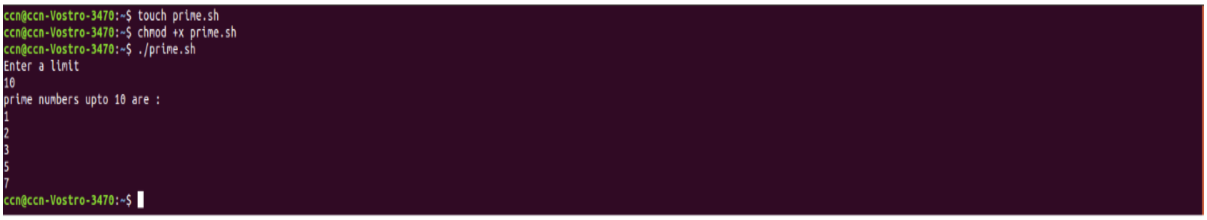
6. Write a script to count the prime numbers in specific range

SOURCE CODE

```
#!/bin/bash
echo "Enter a limit"
read limit
echo "prime numbers upto $limit are :"
echo "1"
i=2
```

```
while [ $i -le $limit ]
do
 flag=1
 j=2
 while [ $j -lt $i ]
  do
     rem=$(( $i % $j ))
     if [ $rem -eq 0 ]
     then
     flag=0
     break
     fi
     j=$(( $j+1 ))
    done
    if [ $flag -eq 1 ]
   then
   echo "$i"
   fi
   i=$(( $i+1 ))
   done
```

OUTPUT

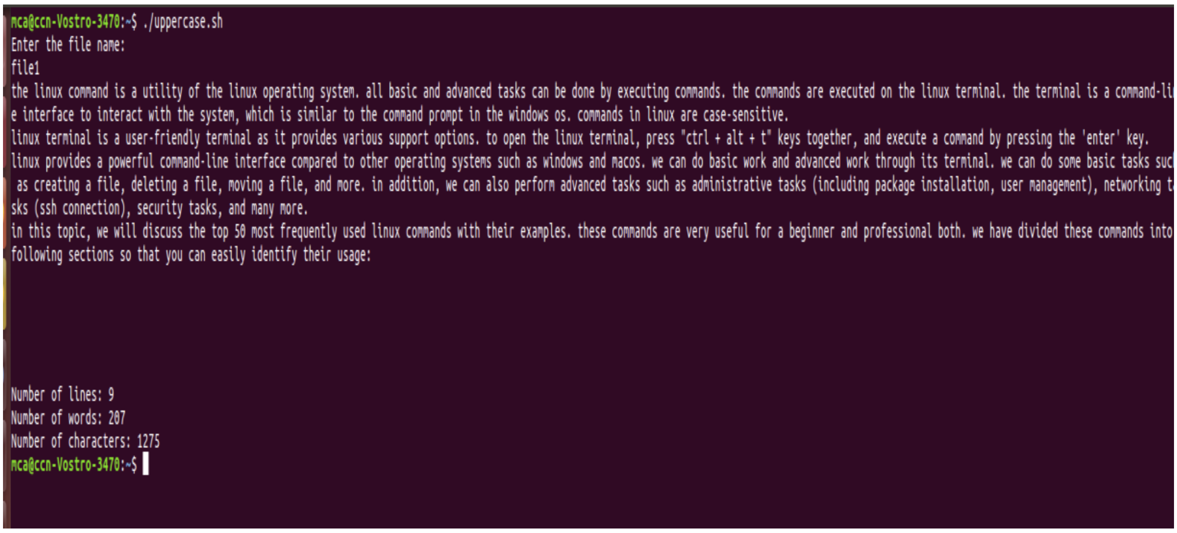


```
ccn@ccn-Vostro-3470:~$ touch prime.sh
ccn@ccn-Vostro-3470:~$ chmod +x prime.sh
ccn@ccn-Vostro-3470:~$ ./prime.sh
Enter a limit
10
prime numbers upto 10 are :
1
2
3
5
7
ccn@ccn-Vostro-3470:~$ 
```

7. Write a script to convert the contents of a given file from uppercase to lowercase and also count the number of lines, words and characters of the resultant file. Also display the resultant file in descending order.

SOURCE CODE

```
#!/bin/bash
echo "Enter the file name:"
read filename
if [ ! -f "$filename" ]; then
  echo "File '$filename' does not exist."
  exit 1
  fi
  cat "$filename" | tr '[:upper:]' '[:lower:]' > lowercase.txt
  lines=$(wc -l lowercase.txt | cut -d ' ' -f 1)
  words=$(wc -w lowercase.txt | cut -d ' ' -f 1)
  characters=$(wc -c lowercase.txt | cut -d ' ' -f 1)
  sort -r lowercase.txt > sorted.txt
  cat sorted.txt
  echo "Number of lines: $lines"
  echo "Number of words: $words"
  echo "Number of characters: $characters"
```



```
mca@ccn-Vostro-3470:~$ ./uppercase.sh
Enter the file name:
file1
the linux command is a utility of the linux operating system. all basic and advanced tasks can be done by executing commands. the commands are executed on the linux terminal. the terminal is a command-li
e interface to interact with the system, which is similar to the command prompt in the windows os. commands in linux are case-sensitive.
linux terminal is a user-friendly terminal as it provides various support options. to open the linux terminal, press "ctrl + alt + t" keys together, and execute a command by pressing the 'enter' key.
linux provides a powerful command-line interface compared to other operating systems such as windows and macos. we can do basic work and advanced work through its terminal. we can do some basic tasks suc
 as creating a file, deleting a file, moving a file, and more. in addition, we can also perform advanced tasks such as administrative tasks (including package installation, user management), networking t
sks (ssh connection), security tasks, and many more.
in this topic, we will discuss the top 50 most frequently used linux commands with their examples. these commands are very useful for a beginner and professional both. we have divided these commands into
following sections so that you can easily identify their usage:






Number of lines: 9
Number of words: 207
Number of characters: 1275
mca@ccn-Vostro-3470:~$ 
```

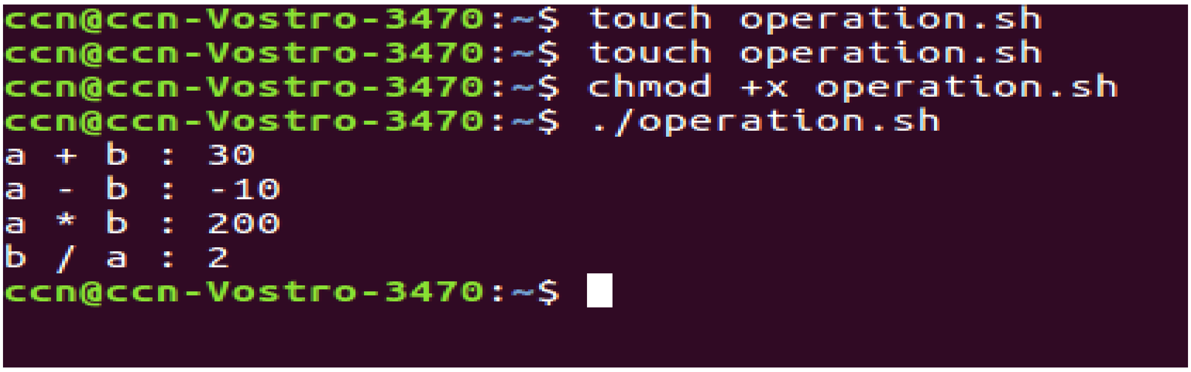
8. Write a script to perform following basic math operation as:

Addition, subtraction, multiplication, division

SOURCE CODE

```
#!/bin/sh
a=10
b=20
val=`expr $a + $b`
echo "a + b : $val"
val=`expr $a - $b`
echo "a - b : $val"
val=`expr $a \* $b`
echo "a * b : $val"
val=`expr $b / $a`
echo "b / a : $val"
```

OUTPUT



```
ccn@ccn-Vostro-3470:~$ touch operation.sh
ccn@ccn-Vostro-3470:~$ touch operation.sh
ccn@ccn-Vostro-3470:~$ chmod +x operation.sh
ccn@ccn-Vostro-3470:~$ ./operation.sh
a + b : 30
a - b : -10
a * b : 200
b / a : 2
ccn@ccn-Vostro-3470:~$
```

9. Read 3 marks of a student and find the average. Display the grade of the student based on the average. *(if..then..elif..fi)*

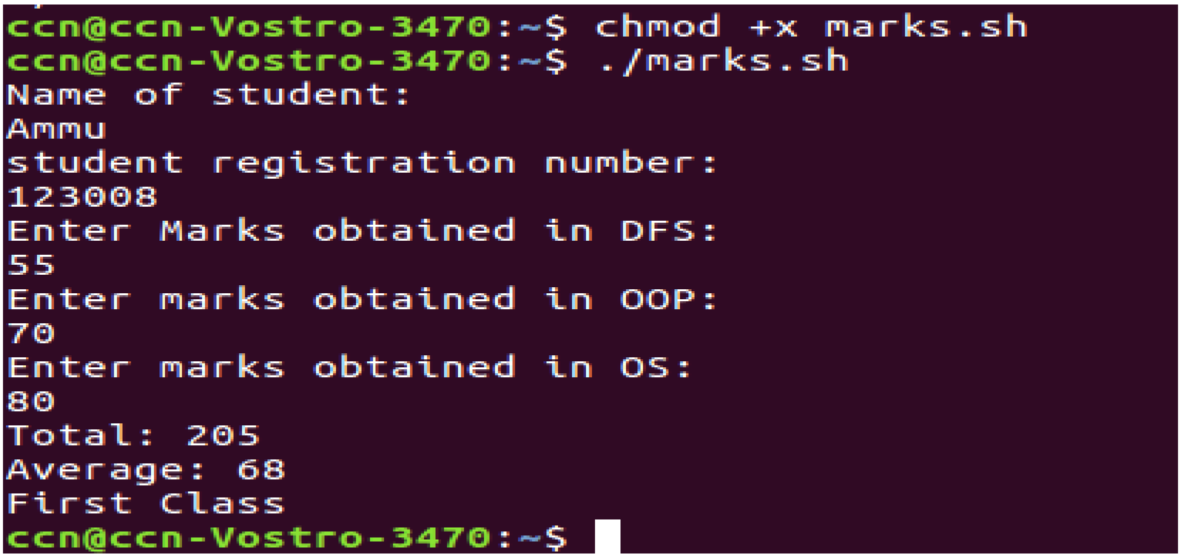
S >= 90%
A < 90%, but >= 80%
B < 80%, but >= 60%
P < 80%, but >= 40%
F < 40%

SOURCE CODE

```
echo "Name of student:"
read name
echo "student registration number:"
read student registration number
echo "Enter Marks obtained in DFS: "
read m1
echo "Enter marks obtained in OOP: "
read m2
```

```
echo "Enter marks obtained in OS: "
read m3
total=`expr $m1 + $m2 + $m3`
avg=`expr $total / 3`
echo "Total: $total"
echo "Average: $avg"
if [ $avg -ge 90 ]
then
echo "Distinction"
elif [ $avg -le 90 ] && [ $avg -ge 50 ]
then
echo "First Class"
elif  [ $avg -le 80 ] && [ $avg -ge 60 ]
then
echo "Second Class"
elif [ $avg -le 80 ] && [ $avg -ge 40 ]
then
echo "Third Class"
else
echo "Fail"
fi
```

OUTPUT



```
ccn@ccn-Vostro-3470:~$ chmod +x marks.sh
ccn@ccn-Vostro-3470:~$ ./marks.sh
Name of student:
Ammu
student registration number:
123008
Enter Marks obtained in DFS:
55
Enter marks obtained in OOP:
70
Enter marks obtained in OS:
80
Total: 205
Average: 68
First Class
ccn@ccn-Vostro-3470:~$
```

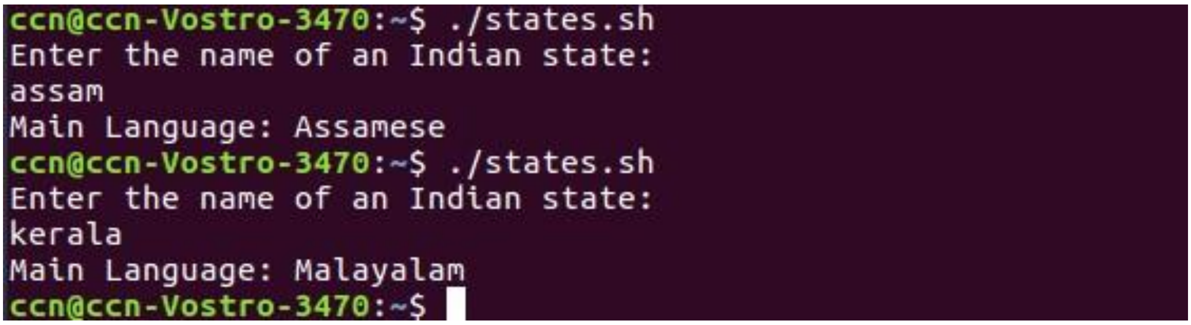
10. Read the name of an Indian state and display the main language according to the table. For other states, the output may be “Unknown”. Use “|” to separate states with same language *(case..esac)*

State	Main Language
Andhra Pradesh	Telugu
Assam	Assamese
Bihar	Hindi
Himachal Pradesh	Hindi
Karnataka	Kannada
Kerala	Malayalam
Lakshadweep	Malayalam
Tamil Nadu	Tamil

SOURCE CODE

```
#!/bin/bash
echo "Enter the name of an Indian state:"
read -r state
state=$(echo "$state" | tr '[:upper:]' '[:lower:]')
echo $state
case $state in "andhra pradesh")
echo "Main Language: Telugu" ;;
"assam")
echo "Main Language: Assamese";;
"bihar")
 echo "Main Language: Hindi";;
"himachal pradesh")
  echo "Main Language: Hindi" ;;
 "karnataka")
  echo "Main Language: Kannada" ;;
 "kerala" | "lakshadweep")
  echo "Main Language: Malayalam"
  "tamil nadu")
  echo "Main Language: Tamil" ;; *)
  echo "Main Language: Unknown";;
  esac
```

OUTPUT



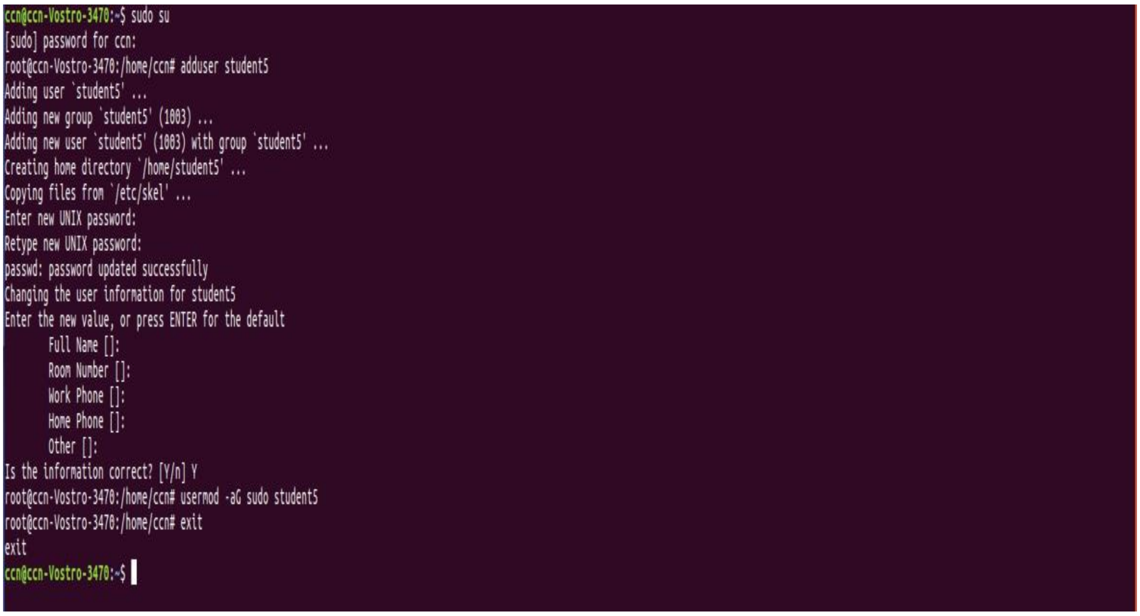
```
ccn@ccn-Vostro-3470:~$ ./states.sh
Enter the name of an Indian state:
assam
Main Language: Assamese
ccn@ccn-Vostro-3470:~$ ./states.sh
Enter the name of an Indian state:
kerala
Main Language: Malayalam
ccn@ccn-Vostro-3470:~$ 
```

11.Change the home folder of all users whose name start with stud from /home/username to /usr/username. Also change the password of username to username123 (e.g., /home/stud25 changes to /usr/stud25 and his/her password changes to stud25123) - *(Use for .. in)*

SOURCE CODE

```
#!/bin/bash
for username in /home/stud*; do
username=$(basename "$username")
new_home="/usr/$username"
sudo usermod -m -d "$new_home" "$username"
new_password="${username}123"
echo -e "$new_password\n$new_password" | sudo passwd "$username"
done
```

OUTPUT



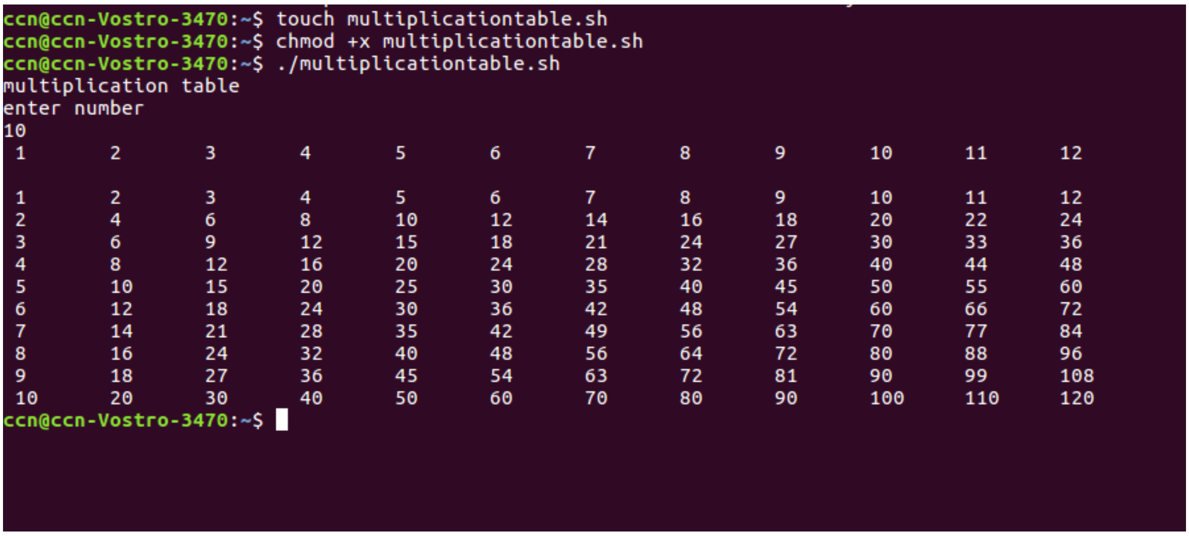
```
ccn@ccn-Vostro-3470:~$ sudo su
[sudo] password for ccn:
root@ccn-Vostro-3470:/home/ccn# adduser student5
Adding user `student5' ...
Adding new group `student5' (1003) ...
Adding new user `student5' (1003) with group `student5' ...
Creating home directory `/home/student5' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for student5
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@ccn-Vostro-3470:/home/ccn# usermod -aG sudo student5
root@ccn-Vostro-3470:/home/ccn# exit
exit
ccn@ccn-Vostro-3470:~$ 
```

12.Read a number and display the multiplication table of the number up to 10 lines. – (Use for((..)))

SOURCE CODE

```
 #!/bin/bash
 echo "multiplication table"
 echo "enter number"
 read n
 m=0
 for((j=1;j<=12;j++))
 do
 echo -n -e " $j\t"
 done
 echo ""
echo
for((i=1;i<=n;i++))
do
for((k=1;k<=12;k++))
do
m=` expr $k \* $i `
echo -n -e " $m\t"
done
echo ""
done
```

OUTPUT



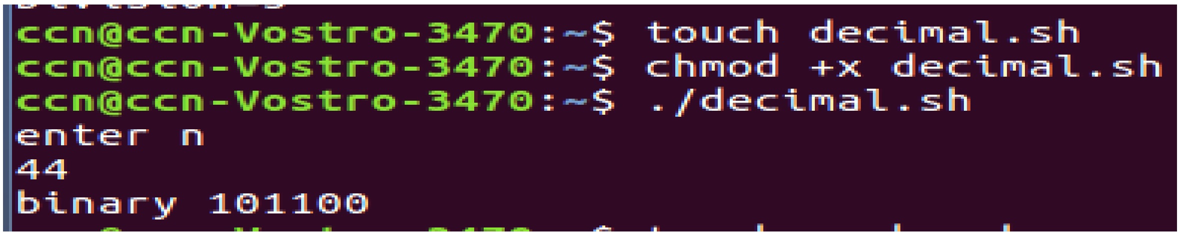
```
ccn@ccn-Vostro-3470:~$ touch multiplicationtable.sh
ccn@ccn-Vostro-3470:~$ chmod +x multiplicationtable.sh
ccn@ccn-Vostro-3470:~$ ./multiplicationtable.sh
multiplication table
enter number
10
 1       2       3       4       5       6       7       8       9       10      11      12

 1       2       3       4       5       6       7       8       9       10      11      12
 2       4       6       8       10      12      14      16      18      20      22      24
 3       6       9       12      15      18      21      24      27      30      33      36
 4       8       12      16      20      24      28      32      36      40      44      48
 5       10      15      20      25      30      35      40      45      50      55      60
 6       12      18      24      30      36      42      48      54      60      66      72
 7       14      21      28      35      42      49      56      63      70      77      84
 8       16      24      32      40      48      56      64      72      80      88      96
 9       18      27      36      45      54      63      72      81      90      99      108
 10      20      30      40      50      60      70      80      90      100     110     120
ccn@ccn-Vostro-3470:~$ 
```

13.Read a Decimal number. Convert it to Binary and display the result. -
(Use while)

SOURCE CODE

```
#!/bin/bash
echo "Enter a decimal number: "
read number
binary_number=""
while [ "$number" -gt 0 ]; do
binary_number="$binary_number$((number % 2))"
number=$((number / 2))
done
echo "The binary number is: $binary_number"
```

OUTPUT



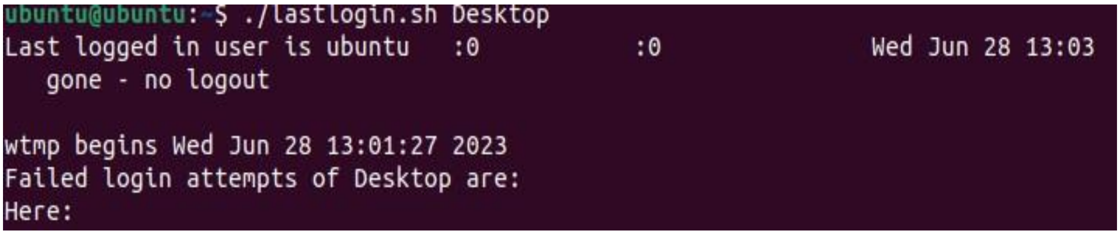
```
ccn@ccn-Vostro-3470:~$ touch decimal.sh
ccn@ccn-Vostro-3470:~$ chmod +x decimal.sh
ccn@ccn-Vostro-3470:~$ ./decimal.sh
enter n
44
binary 101100
```

14.Look at the system log files. Write a shell script to extract the last login details of a particular user and list out all failed logins. Store the results to a file. The user name should be given as a command line argument.

SOURCE CODE

```
#! /bin/bash
if [ $# -eq 0 ]
then
        echo "Please try again with a valid argument";
        exit
  fi
        lastLogin=$(last -n 1);
        echo "Last logged in user is $lastLogin"
        loginAttempts=$(sudo cat /var/log/auth.log | grep $1 | grep failed)
        echo "Failed login attempts of $1 are:"
        echo "Here: $loginAttempts"
```

OUTPUT



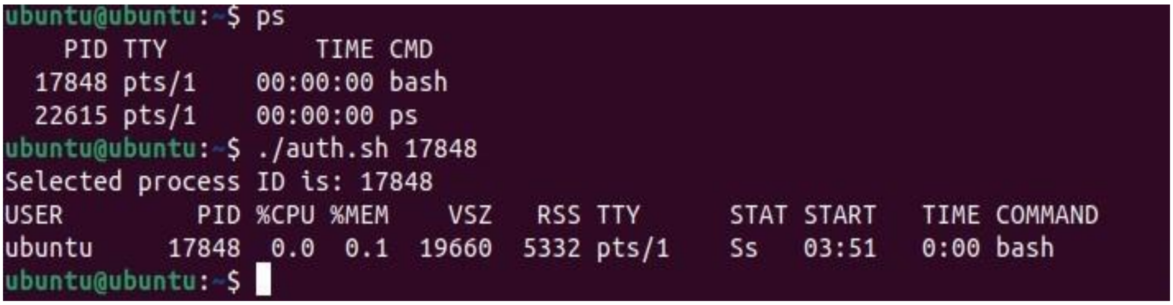
```
ubuntu@ubuntu:~$ ./lastlogin.sh Desktop
Last logged in user is ubuntu   :0          :0               Wed Jun 28 13:03
   gone - no logout

wtmp begins Wed Jun 28 13:01:27 2023
Failed login attempts of Desktop are:
Here:
```

15.Write a shell script to display the details of a particular process currently running. Assume that you have necessary permissions. The process name/id is to be given as a command line argument.

SOURCE CODE

```
#! /bin/bash
if [ $# -eq 0 ]
then
      echo "Please try again with a valid argument";
      exit
fi
      echo "Selected process ID is: $1"
      ps -q $1 -axu
```

OUTPUT



```
ubuntu@ubuntu:~$ ps
    PID TTY          TIME CMD
  17848 pts/1    00:00:00 bash
  22615 pts/1    00:00:00 ps
ubuntu@ubuntu:~$ ./auth.sh 17848
Selected process ID is: 17848
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
ubuntu     17848  0.0  0.1  19660  5332 pts/1    Ss   03:51   0:00 bash
ubuntu@ubuntu:~$ 
```