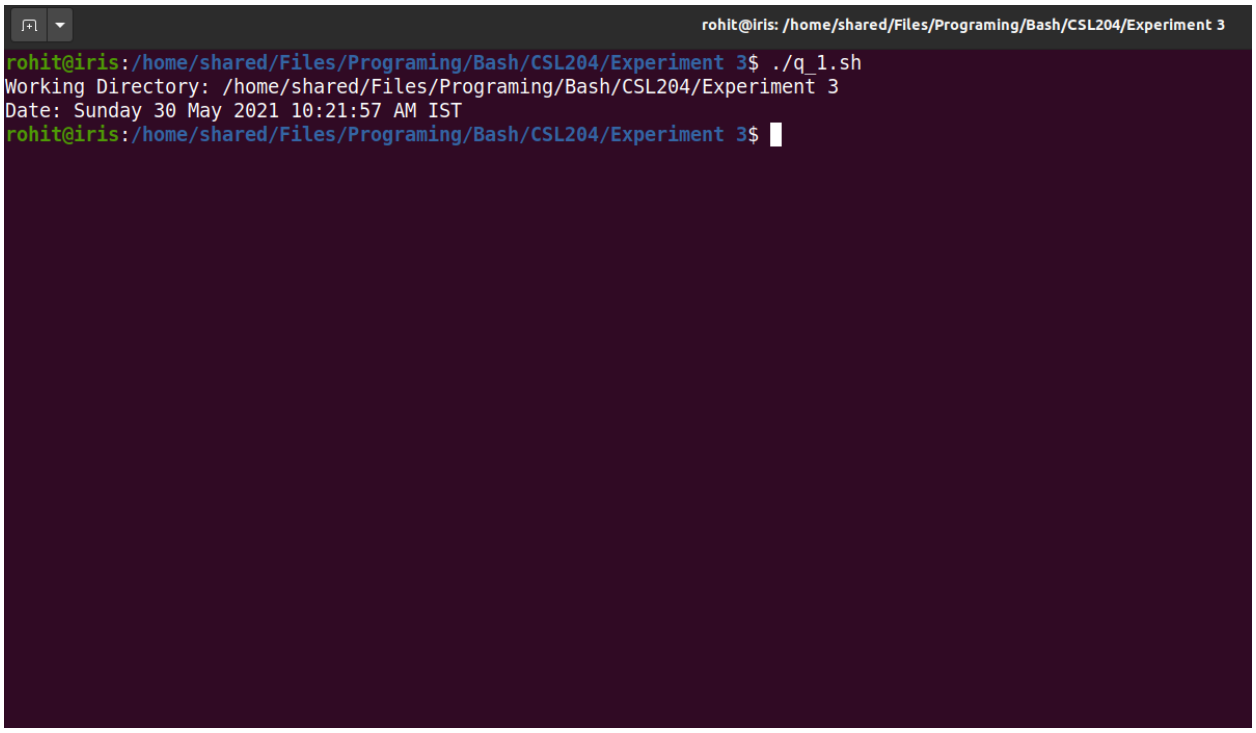
Experiment 3
Bash Shell Scripting

Done By: Rohit Karunakarn
Date of Submission: 31-05-2021

Program 1: Write a shell script program to display the name of the current working directory and date

```
#! /bin/bash

echo -e "Working Directory: $(pwd)\nDate: $(date)"
```

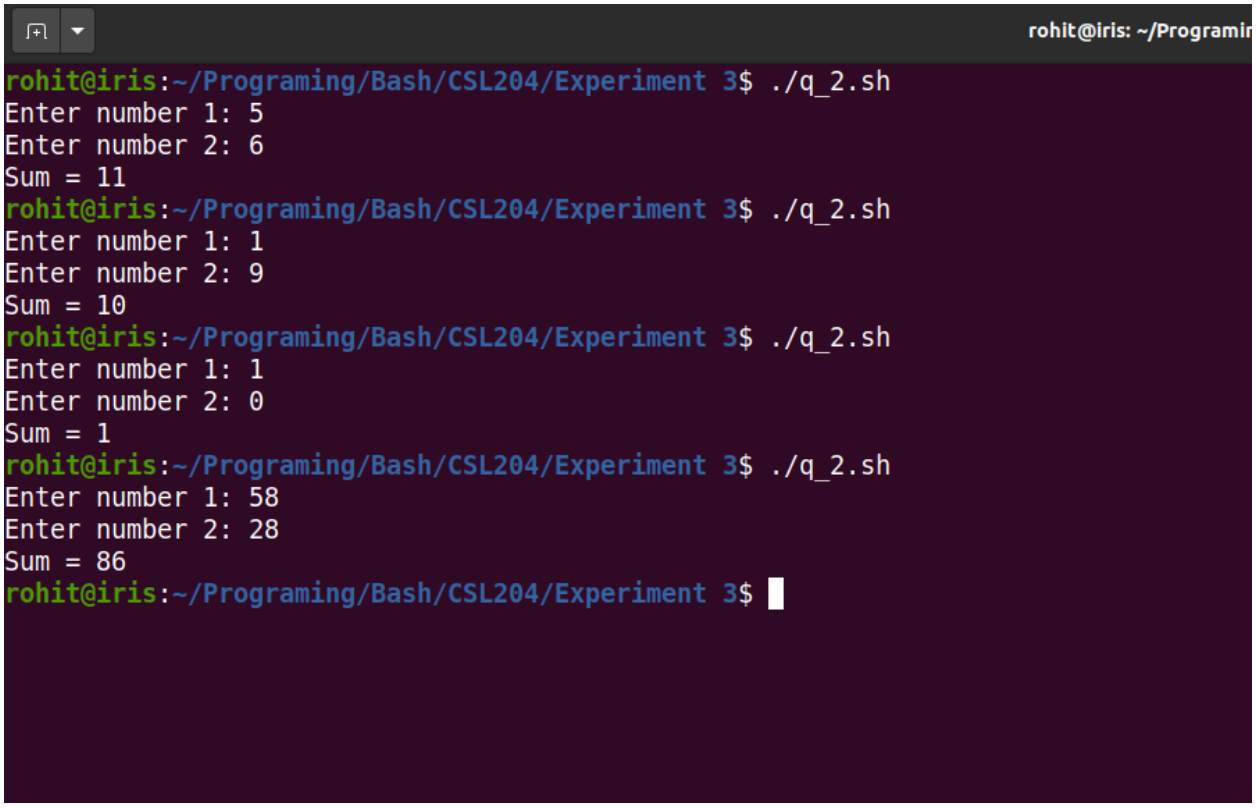
Screen shots:



Program 2: Write a shell script program to perform arithmetic operation on two numbers.

```
#! /bin/bash

echo -n "Enter number 1: "
read a
echo -n "Enter number 2: "
read b
let "c = $a+$b"
echo "Sum = $c"
```

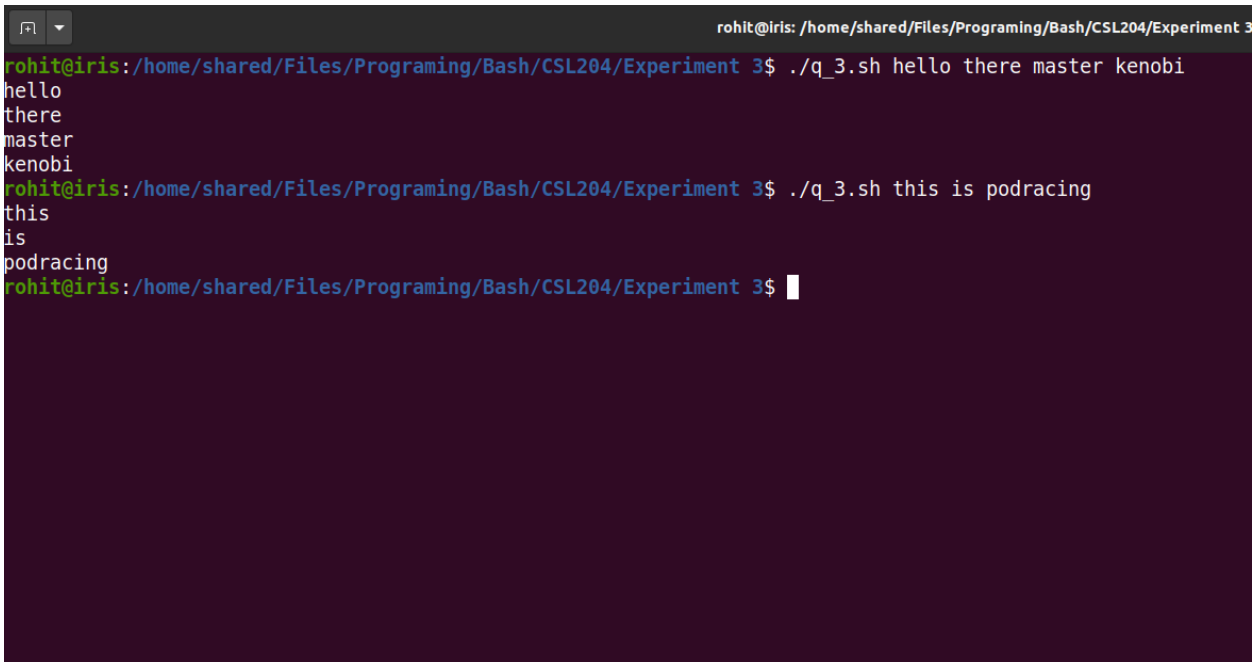
Screen shots:



Program 3: Write a shell script program to pass arguments to the program and display the count of arguments and content.

```
#! /bin/bash

for var in $@
do
        echo "${var}"
done
```

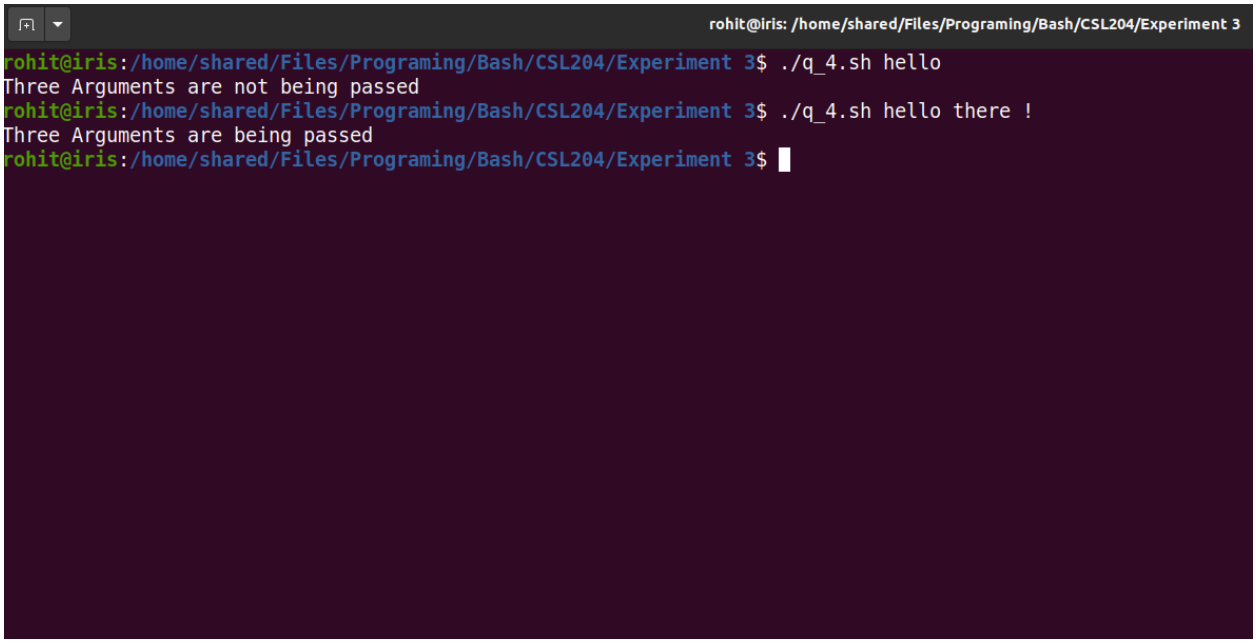
Screen shots:



Program 4: Write a shell script program to test whether 3 arguments are being passed to it.

```
#! /bin/bash

if [ $# -eq 3 ] ; then
        echo "Three Arguments are being passed"
else
        echo "Three Arguments are not being passed"
fi
```

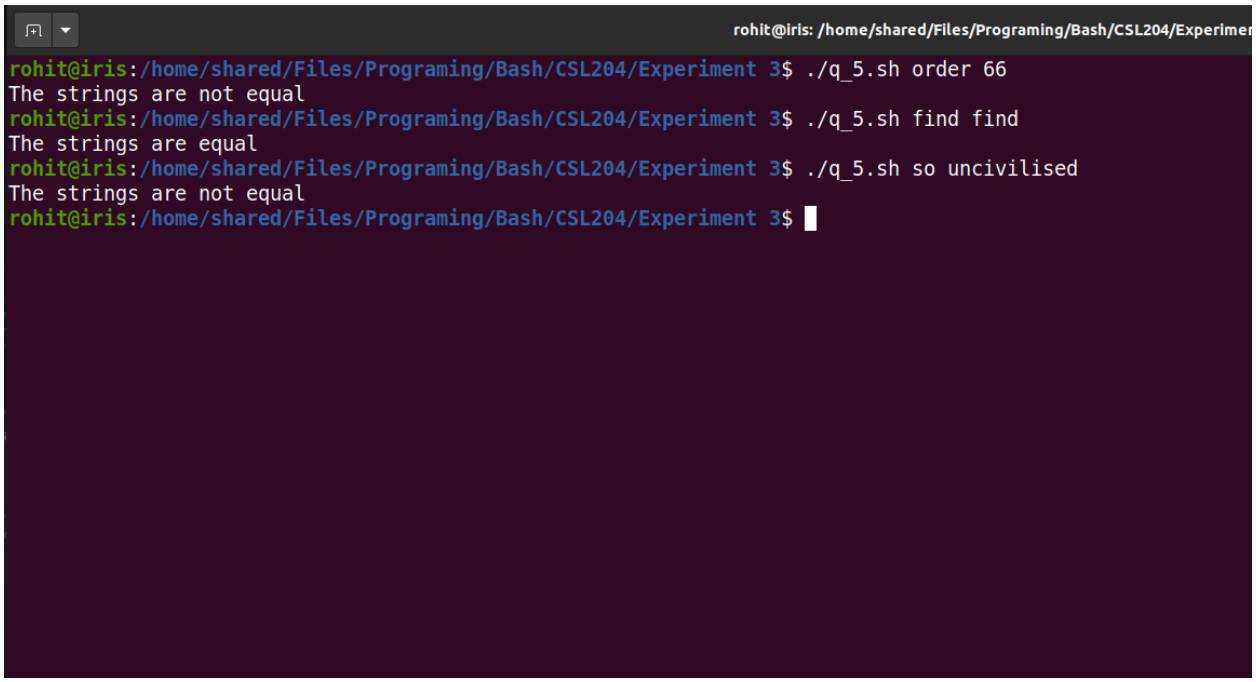
Screen shots:



Program 5: Write a shell script program to check whether two strings sent as command line arguments are same or not using test command.

```
#! /bin/bash

if [ ! $# -eq 2 ] ; then
        echo "Two arguments are needed"
else
        if [ $1 = $2 ] ; then
        echo "The strings are equal"
        else
        echo "The strings are not equal"
        fi
fi
```

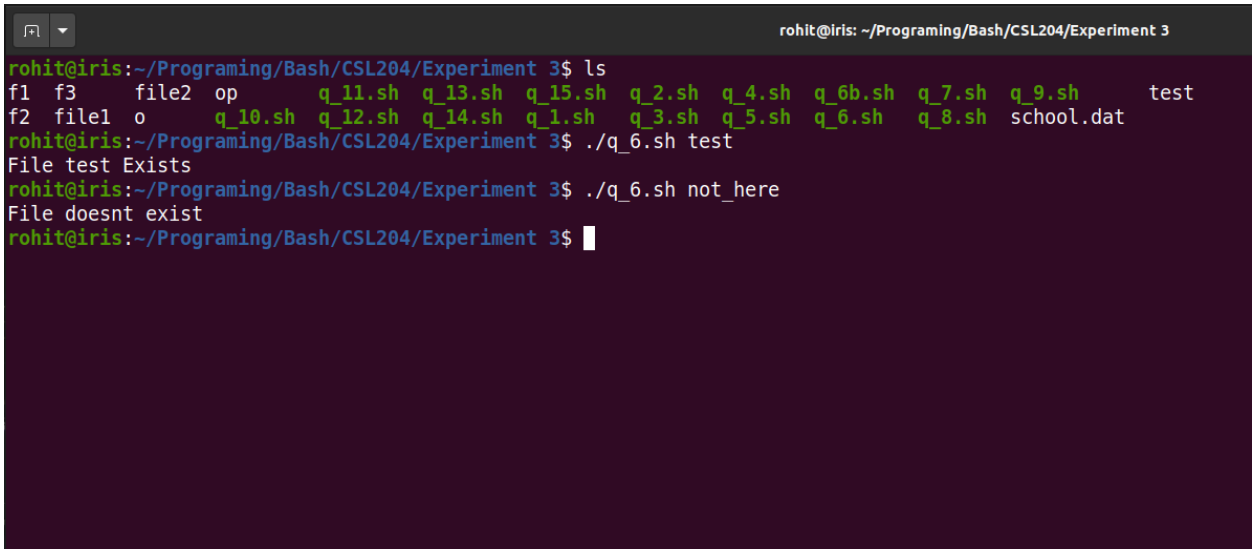
Screen shots:



Program 6.a Let argument be a filename. Write a shell script program to check whether the file exists or not.

```
#! /bin/bash

if [ $# -eq 0 ]
then
        echo "Atleast one Argument is required"
else
        if [ -f $1 ]; then
        echo "File ${1} Exists"
        else
        echo "File doesnt exist"
        fi
fi
```

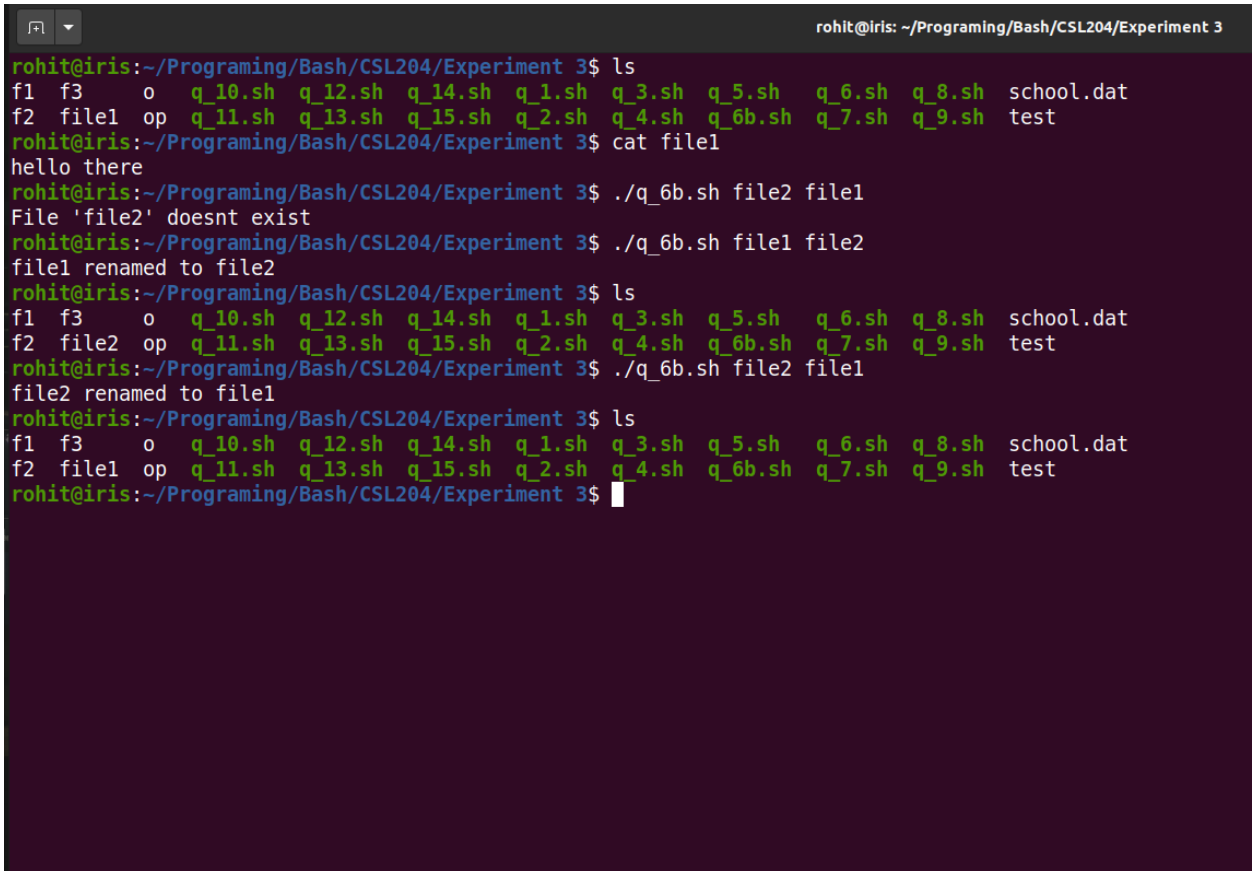
Screen shots:



Program 6.b: Modify the above program to rename a filename with another name passed as argument, Also check number of parameters being passed.

```
#! /bin/bash

if [ $# -lt 2 ]
then
        echo "Atleast two Arguments is required"
else
        if [ -f $1 ]; then
        mv $1 $2
        echo "$1 renamed to $2"
        else
        echo "File '$1' doesnt exist"
        fi
fi
```

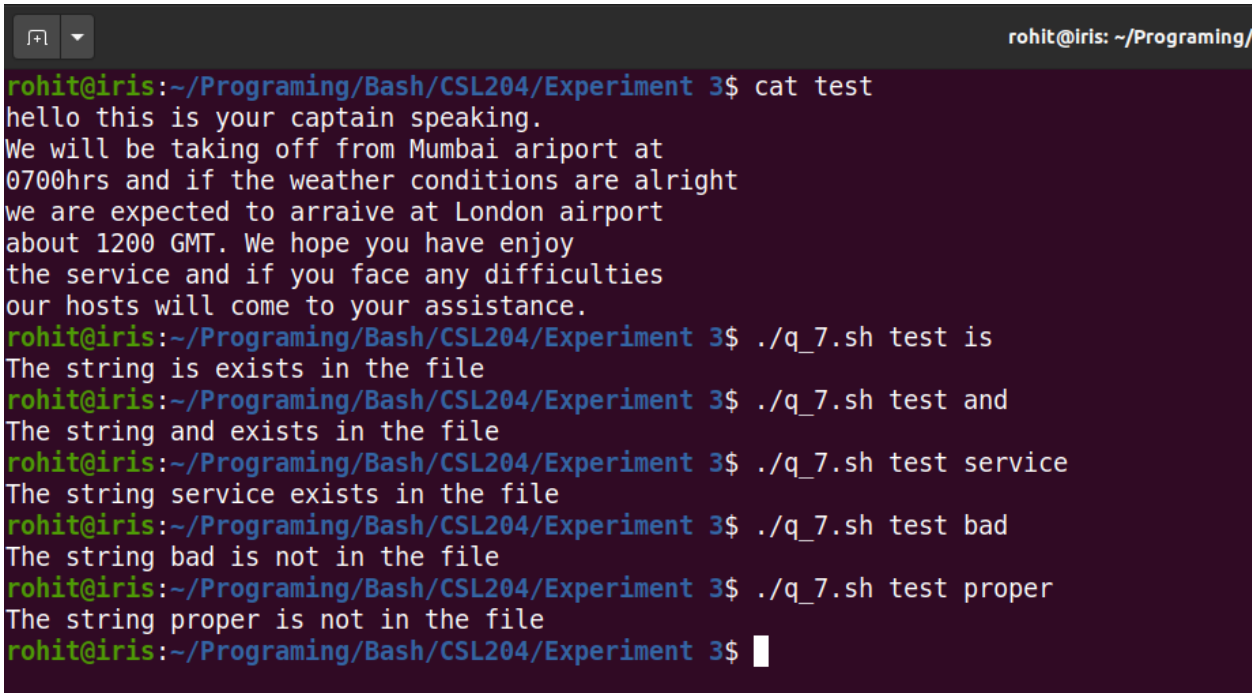
Screen shots:



Program 7: Write a shell script program to test whether a string is present in a file or not

```
#! /bin/bash

if [ $# -lt 2 ]
then
        echo "Atleast two Argument is required"
        echo "./q_7.sh [file] [string to be searched]"
else
        if [ -f $1 ]; then
        found=$(grep $2 $1)
        if [ -n "$found" ]
        then
        echo "The string $2 exists in the file"
        else
        echo "The string $2 is not in the file"
        fi
        else
        echo "File doesnt exist"
        fi
fi
```

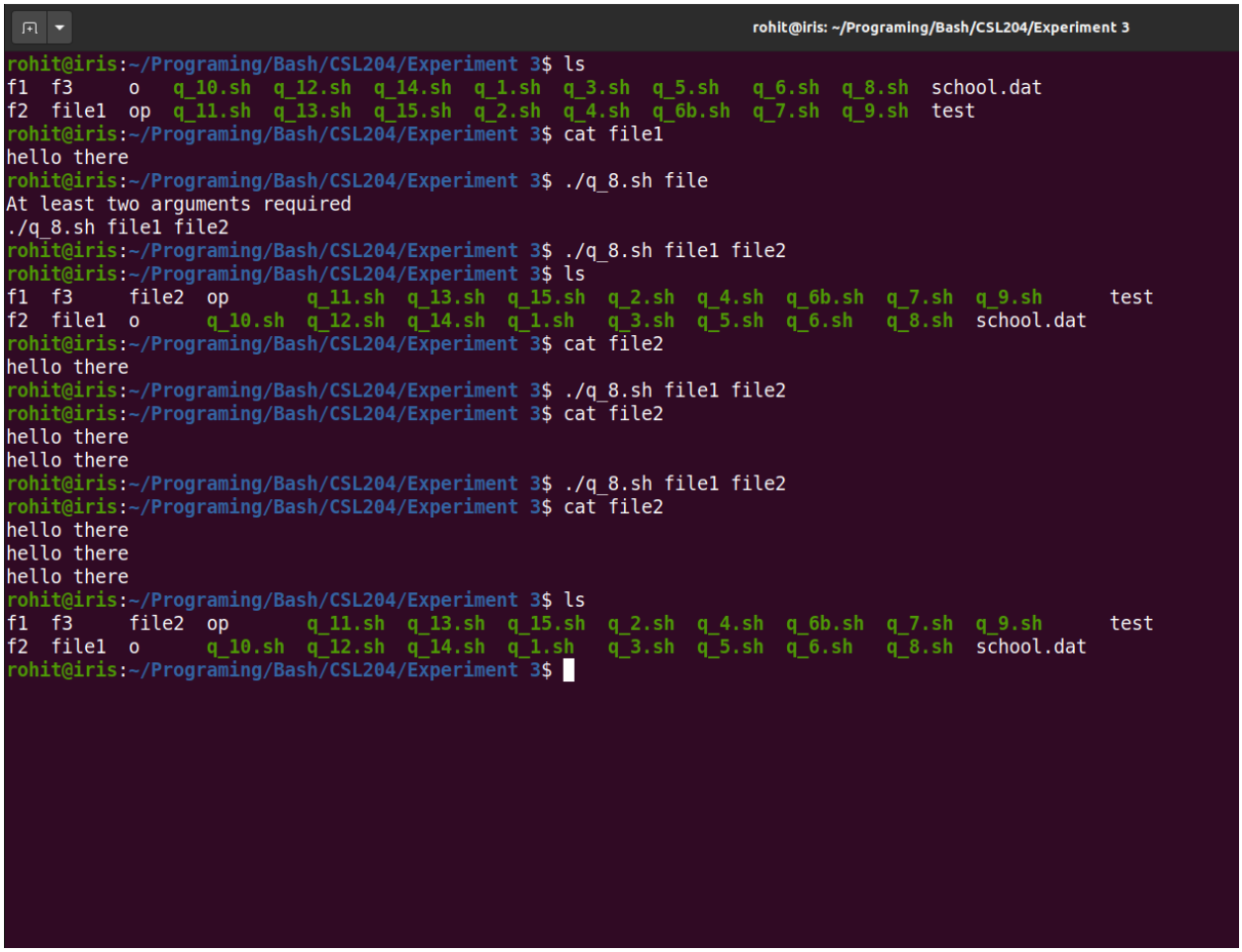
Screen shots:



Program 8: Write a shell script program to copy content of file1 to file 2. If file2 exists then append the content of file1 to its original content.

```
#! /bin/bash

if [ $# -lt 2 ]
then
        echo "At least two arguments required"
        echo "$0 file1 file2"
else
        if [ -f $2 ]; then
        cat $1 >> $2
        else
        cat $1 > $2
        fi
fi
```

Screen shots:



Program 9: Write a shell script program that take first 10 lines from first file and rest 10lines from second file and move them in a third file. The filename will be sent as command line arguments.

```
#! /bin/bash

if [ $# -lt 3 ]
then
        echo "Enter three file names"
else
        echo -ne "" > $3

        let i=0
        while  IFS='\n' read -r line
        do
                if [ $i -eq 10 ]
                then
                        break
                fi
                echo $line >> "$3"
                let "i=$i+1"
        done < $1

        let i=0
        while  IFS='\n' read -r line
        do
                if [ $i -eq 10 ]
                then
                        break
                fi
                echo $line >> "$3"
                let "i=$i+1"
        done < $2

fi
```

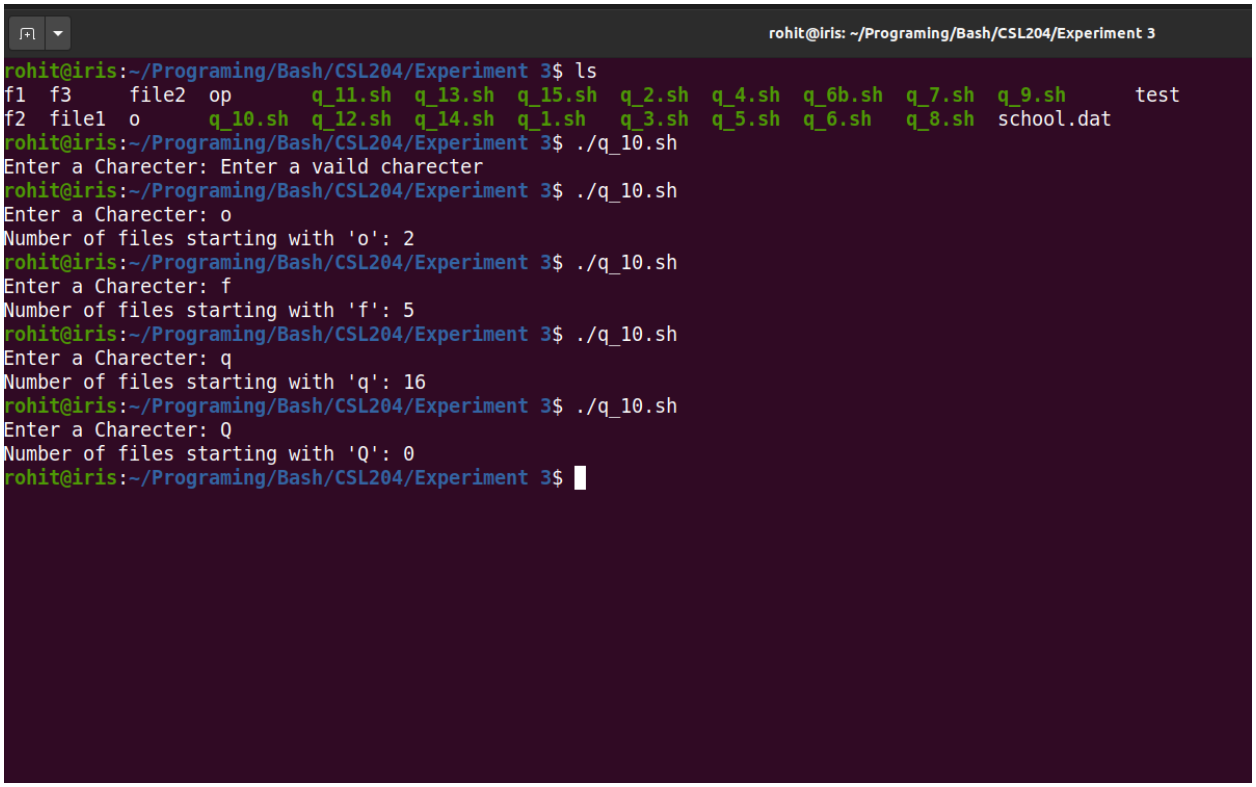
Screen shots:

```
rohit@iris:~/Programing/Bash/CSL204/Experiment 3$ cat f1
Hello I am file one
and my purpose is to be an
argument for question 9
but I will continue to exist
even after my purose is served
i just need 10 lines
which will be in file 3, eventually.
A word about file 2, he is not as
civil as me, anyway here we are
line 10 of file 1
these lines are just to make
sure that only the first ten lines
get copied into file3
i pity them, their existance is
almost as futile as mine,
but we together serve a greater purpose
rohit@iris:~/Programing/Bash/CSL204/Experiment 3$ cat f2
I am file two
and why am i the second one??
I am as capable as that file1
and why did you pass me as the
second argument, you are just incorrigible.
You are as vile as that file one
I shall exist even after the command and
what are you going to do about that
I don't think you deserve
line 10 of file 2
or these lines, I won't even
give these to file three no matter what
rohit@iris:~/Programing/Bash/CSL204/Experiment 3$ ./q_9 f1 f2 f3
bash: ./q_9: No such file or directory
rohit@iris:~/Programing/Bash/CSL204/Experiment 3$ ./q_9.sh f1 f2 f3
rohit@iris:~/Programing/Bash/CSL204/Experiment 3$ cat f3
Hello I am file one
and my purpose is to be an
argument for question 9
but I will continue to exist
even after my purose is served
i just need 10 lines
which will be in file 3, eventually.
A word about file 2, he is not as
civil as me, anyway here we are
line 10 of file 1
I am file two
and why am i the second one??
I am as capable as that file1
and why did you pass me as the
second argument, you are just incorrigible.
You are as vile as that file one
I shall exist even after the command and
what are you going to do about that
I don't think you deserve
line 10 of file 2
rohit@iris:~/Programing/Bash/CSL204/Experiment 3$
```

Program 10: Write a shell script program to show number of files in the current directory beginning with a specified character.

```
#! /bin/bash

echo -n "Enter a Charecter: "
read -r ch

if [ -z "$ch" ]; then
        echo "Enter a vaild charecter"
else
        echo -ne "Number of files starting with '$ch': "
        ls -dF ${ch}* 2>/dev/null | grep -E '[^@|^/]$' | wc -l
fi
```

Screen shots:



Program 11: Write a shell script program to read a line from a file and store them into another file after converting all the vowels from first file into uppercase.

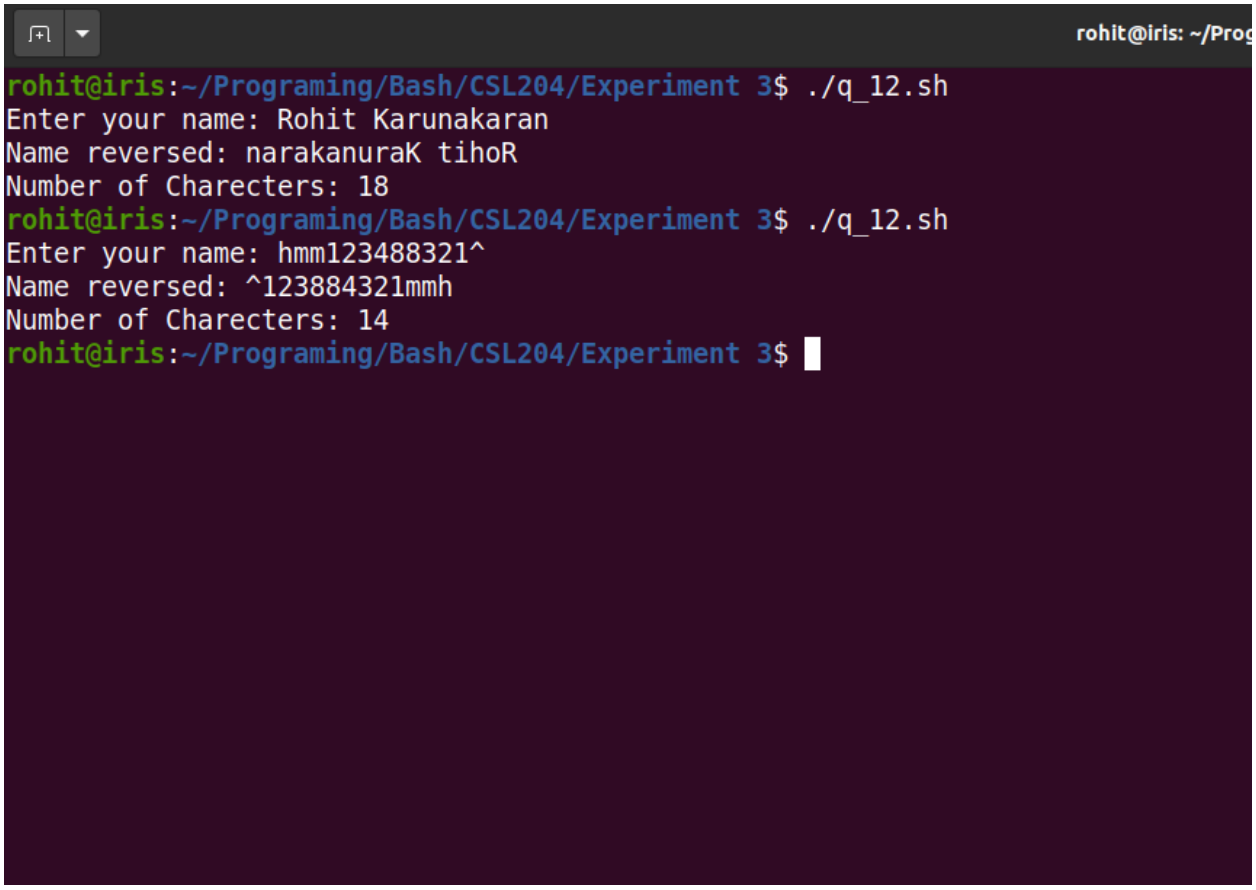
```
#! /bin/bash

if [ $# -lt 1 ]; then
        echo "Pass two file names as command line argument"
else
        read -r l < $1
        echo $l | tr "aeiou" "AEIOU" > $2
fi
```

Screen shots:

Program 12: Write a shell script program that accept the name of the user and prints the entire name in reverse and also print the length of the entire name

```
#! /bin/bash

echo -ne "Enter your name: "
read -r l
echo -ne "Name reversed: "
echo -e $l | rev
echo -ne "Number of Charecters: "
echo -e $l | wc -m
```

Screen shots:



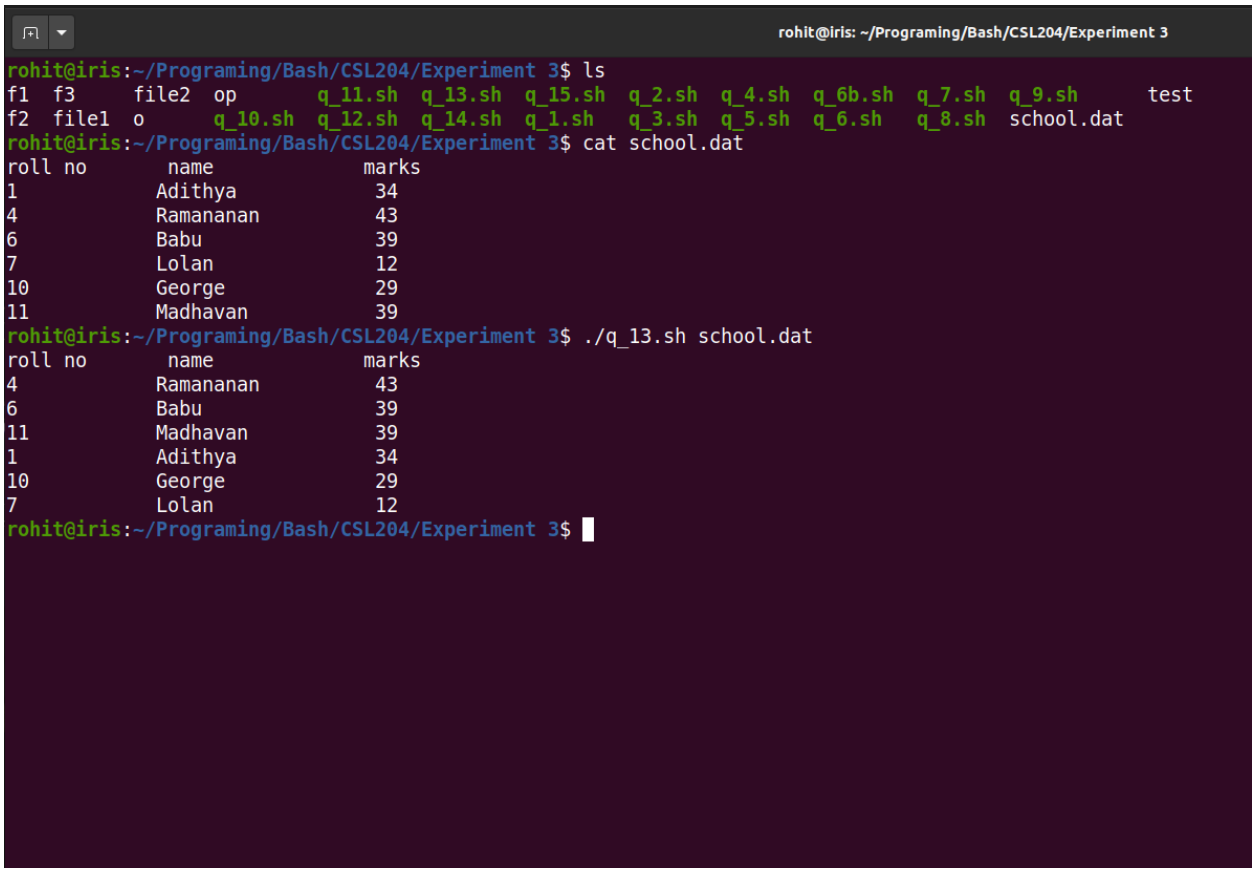
Program 13: Consider a file school.dat with the following fields: roll no, name and marks. Write a shell script program to sort the file in descending order of marks

#! /bin/bash

file="school.dat"

head -n 1 $file
tail -n +2 $file| sort -s -r -nk3

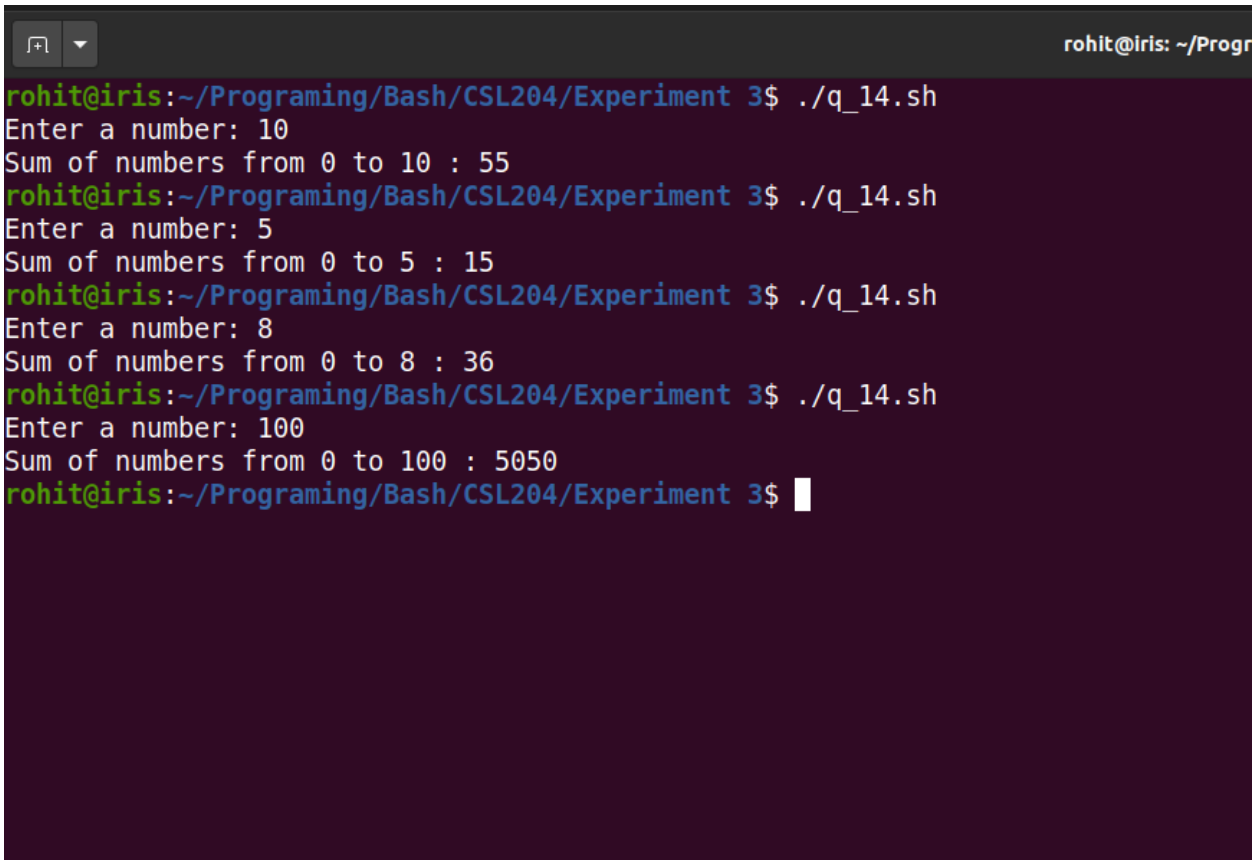
Screen shots:



Program 14: Write a shell script program to display sum of first n numbers

```
#! /bin/bash

echo -n "Enter a number: "
read -r n

#let s="n*(n+1)/2"
let s=0
let i=0

while [ $i -le $n ]
do
        let "s=$s+$i"
        let "i=$i+1"
done

echo "Sum of numbers from 0 to ${n} : ${s}"
```

Screen shots:



Program 15: Menu driven program to display the day when inputting 1 to 7

```
#! /bin/bash

echo "----Day Of The Week----"

let RUN=true
while [ $RUN ]
do
        echo -n "Enter a number between 1 and 7: "
        read -r n

        case $n in
        1)
        echo "Day is Monday";;
        2)
        echo "Day is Tuesday";;
        3)
        echo "Day is Wednesday";;
        4)
        echo "Day is Thursday";;
        5)
        echo "Day is Friday";;
        6)
        echo "Day is Saturday";;
        7)
        echo "Day is Sunday";;
        *)
        echo "Error: Enter a proper input" 1>&2
        esac

        echo -ne "Do you want to continue? (n to exit)"
        read ch
        if [ "$ch" = "n" ] || [ "$ch" = "N" ]; then
        exit 0
        fi
        echo ""

done
```

Screen shots: