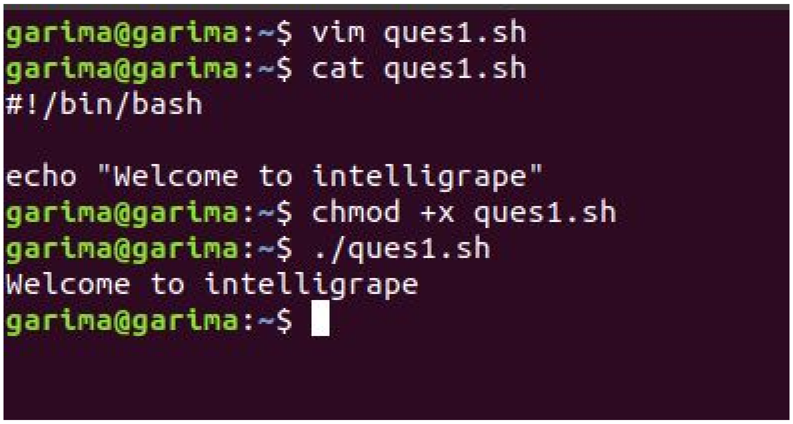
ASSESSMENT - 16

Shell Scripting

TO THE NEW

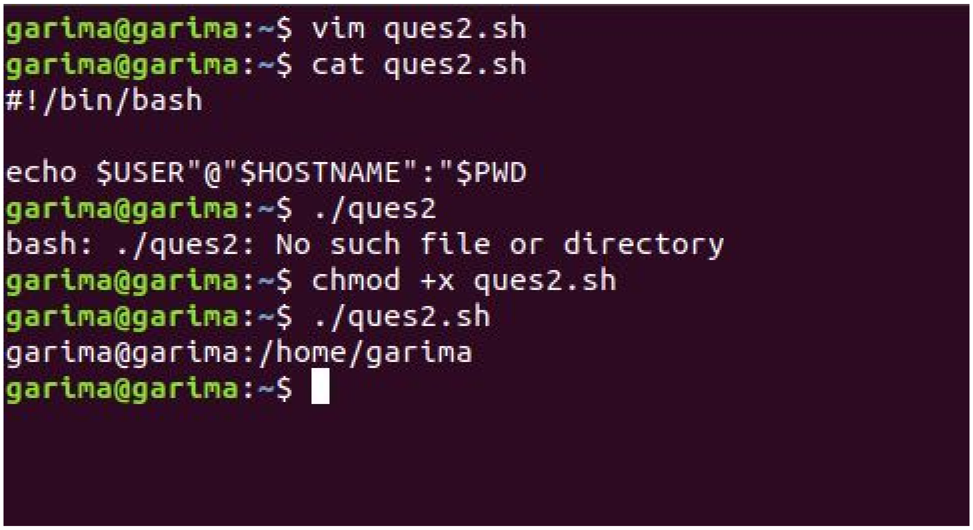
1. (output to terminal)Write a script to print:

a. “Welcome to Intelligrape”



```
garima@garima:~$ vim ques1.sh
garima@garima:~$ cat ques1.sh
#!/bin/bash

echo "Welcome to intelligrape"
garima@garima:~$ chmod +x ques1.sh
garima@garima:~$ ./ques1.sh
Welcome to intelligrape
garima@garima:~$ 
```

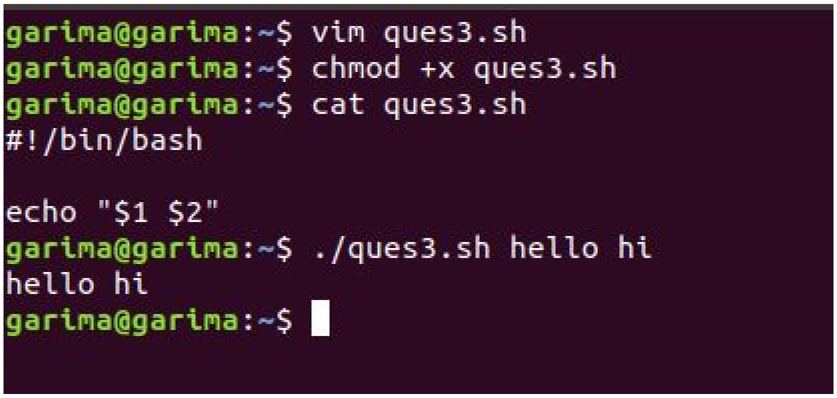
b. <username>@<hostname>:<your present working directory>



```
garima@garima:~$ vim ques2.sh
garima@garima:~$ cat ques2.sh
#!/bin/bash

echo $USER"@"$HOSTNAME":"$PWD
garima@garima:~$ ./ques2
bash: ./ques2: No such file or directory
garima@garima:~$ chmod +x ques2.sh
garima@garima:~$ ./ques2.sh
garima@garima:/home/garima
garima@garima:~$ 
```

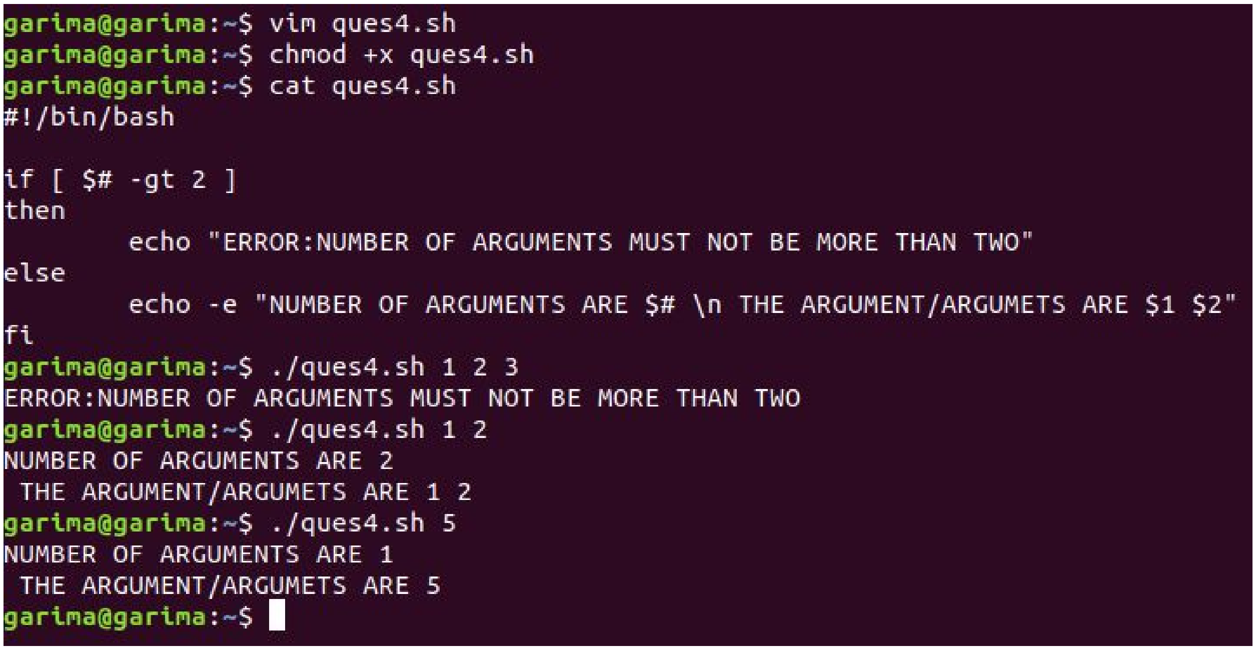
2 (arguments)Write a script

a. which takes in two arguments and print those arguments.



```
garima@garima:~$ vim ques3.sh
garima@garima:~$ chmod +x ques3.sh
garima@garima:~$ cat ques3.sh
#!/bin/bash

echo "$1 $2"
garima@garima:~$ ./ques3.sh hello hi
hello hi
garima@garima:~$ 
```

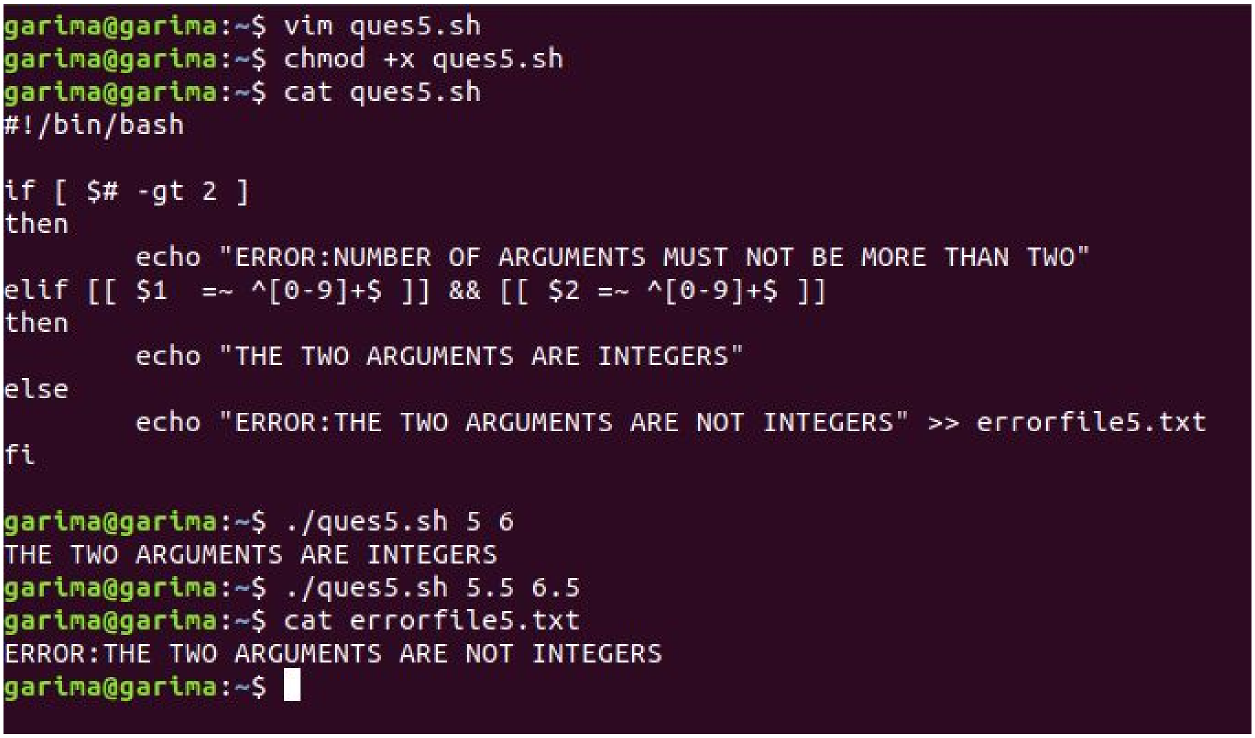
b. which checks the number of arguments passed and if the number is greater than two print ERROR messages along with printing the number of arguments.



```
garima@garima:~$ vim ques4.sh
garima@garima:~$ chmod +x ques4.sh
garima@garima:~$ cat ques4.sh
#!/bin/bash

if [ $# -gt 2 ]
then
        echo "ERROR:NUMBER OF ARGUMENTS MUST NOT BE MORE THAN TWO"
else
        echo -e "NUMBER OF ARGUMENTS ARE $# \n THE ARGUMENT/ARGUMETS ARE $1 $2"
fi
garima@garima:~$ ./ques4.sh 1 2 3
ERROR:NUMBER OF ARGUMENTS MUST NOT BE MORE THAN TWO
garima@garima:~$ ./ques4.sh 1 2
NUMBER OF ARGUMENTS ARE 2
 THE ARGUMENT/ARGUMETS ARE 1 2
garima@garima:~$ ./ques4.sh 5
NUMBER OF ARGUMENTS ARE 1
 THE ARGUMENT/ARGUMETS ARE 5
garima@garima:~$ 
```

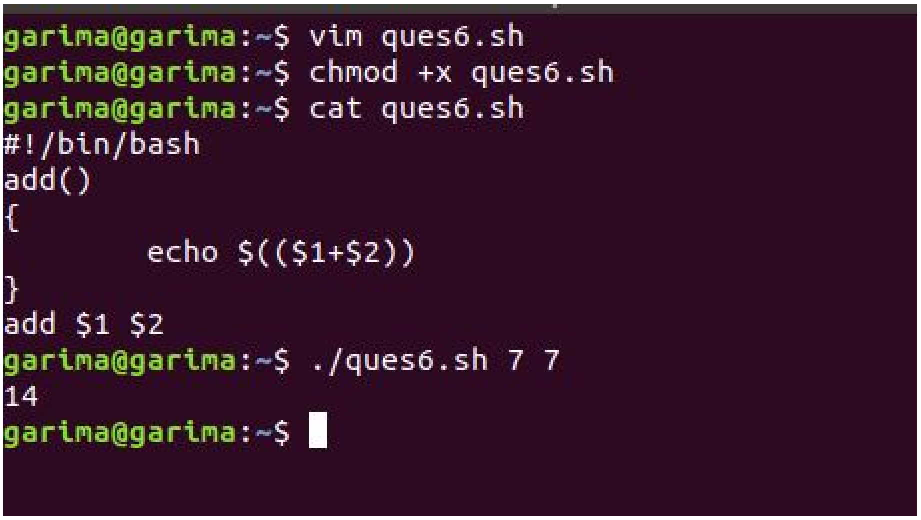
3. Continue with the above script

a. check the two arguments are only integer values and if these are not integers print the proper error on terminal and also log it into a file.



```
garima@garima:~$ vim ques5.sh
garima@garima:~$ chmod +x ques5.sh
garima@garima:~$ cat ques5.sh
#!/bin/bash

if [ $# -gt 2 ]
then
        echo "ERROR:NUMBER OF ARGUMENTS MUST NOT BE MORE THAN TWO"
elif [[ $1  =~ ^[0-9]+$ ]] && [[ $2 =~ ^[0-9]+$ ]]
then
        echo "THE TWO ARGUMENTS ARE INTEGERS"
else
        echo "ERROR:THE TWO ARGUMENTS ARE NOT INTEGERS" >> errorfile5.txt
fi

garima@garima:~$ ./ques5.sh 5 6
THE TWO ARGUMENTS ARE INTEGERS
garima@garima:~$ ./ques5.sh 5.5 6.5
garima@garima:~$ cat errorfile5.txt
ERROR:THE TWO ARGUMENTS ARE NOT INTEGERS
garima@garima:~$ 
```

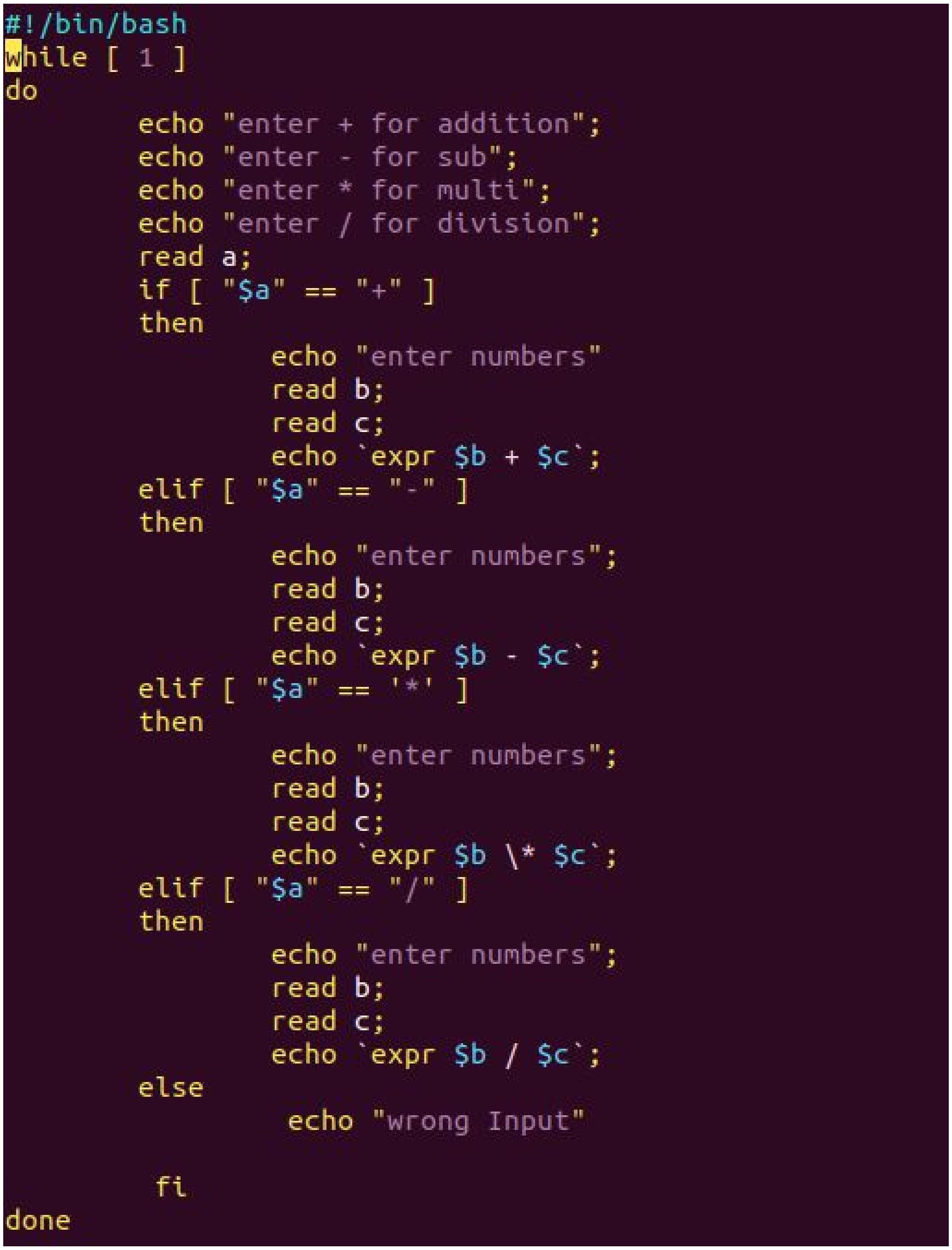
b. perform addition on the two arguments and print the result on screen. Use function for this.



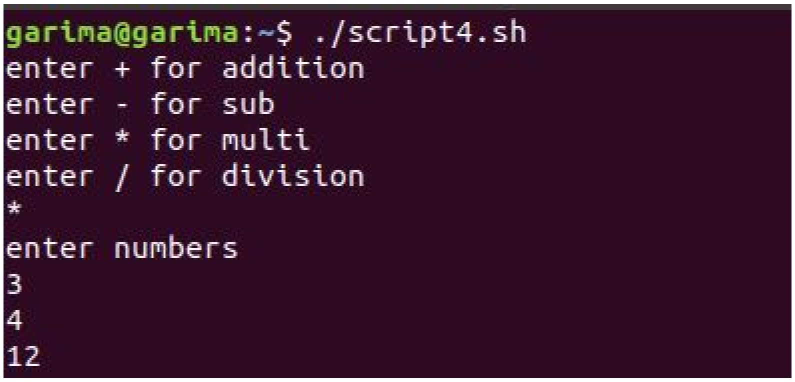
```
garima@garima:~$ vim ques6.sh
garima@garima:~$ chmod +x ques6.sh
garima@garima:~$ cat ques6.sh
#!/bin/bash
add()
{
        echo $(($1+$2))
}
add $1 $2
garima@garima:~$ ./ques6.sh 7 7
14
garima@garima:~$ 
```

4. Create a calculator using the above script which would perform addition,subtraction,division and multiplication.

a. the script should ask user which operation the user wants to perform:+,-,*,/

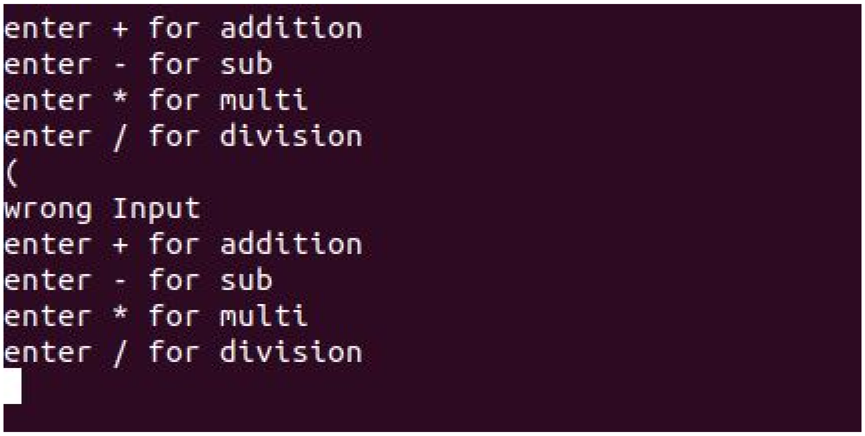


```
#!/bin/bash
while [ 1 ]
do
        echo "enter + for addition";
        echo "enter - for sub";
        echo "enter * for multi";
        echo "enter / for division";
        read a;
        if [ "$a" == "+" ]
        then
                echo "enter numbers"
                read b;
                read c;
                echo `expr $b + $c`;
        elif [ "$a" == "-" ]
        then
                echo "enter numbers";
                read b;
                read c;
                echo `expr $b - $c`;
        elif [ "$a" == '*' ]
        then
                echo "enter numbers";
                read b;
                read c;
                echo `expr $b \* $c`;
        elif [ "$a" == "/" ]
        then
                echo "enter numbers";
                read b;
                read c;
                echo `expr $b / $c`;
        else
                 echo "wrong Input"

         fi
done
```



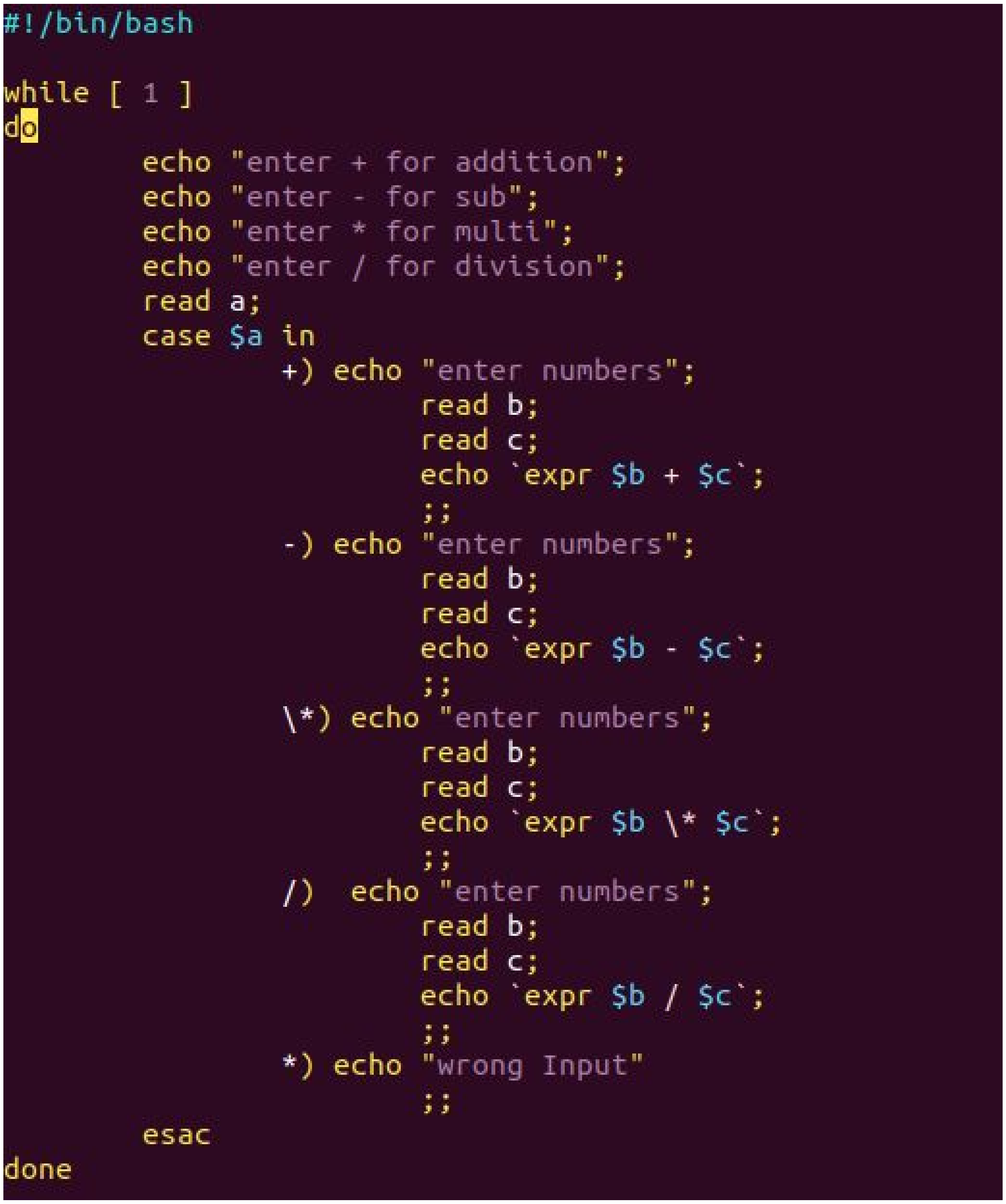
```
garima@garima:~$ ./script4.sh
enter + for addition
enter - for sub
enter * for multi
enter / for division
*
enter numbers
3
4
12
```

b. if user enters other than “+.-,*,/”, print proper messages on the terminal and keep on asking for correct input(use while loop to accomplish this).



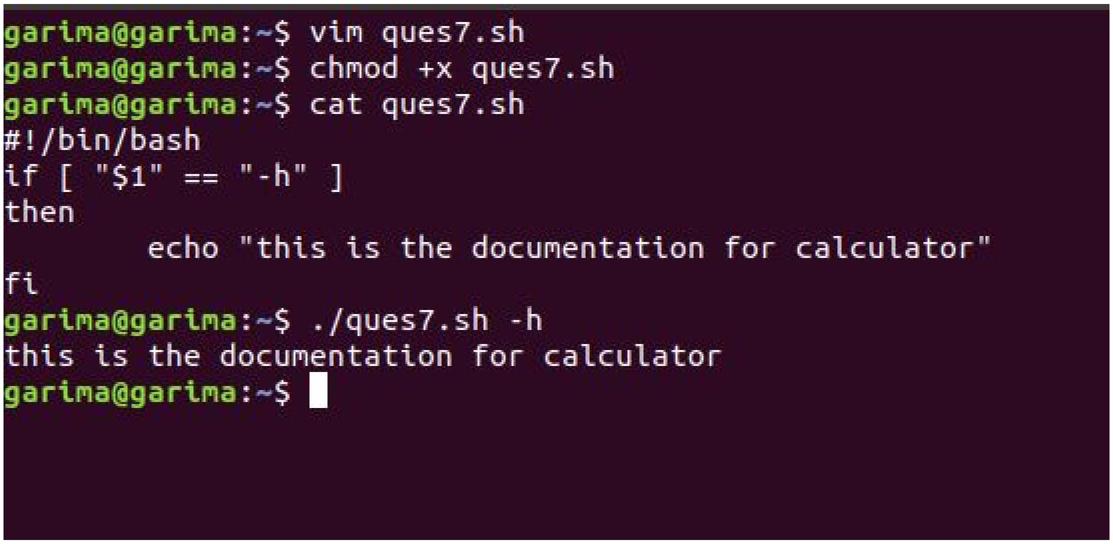
```
enter + for addition
enter - for sub
enter * for multi
enter / for division
(
wrong Input
enter + for addition
enter - for sub
enter * for multi
enter / for division
```

c. Use case statements instead of if.



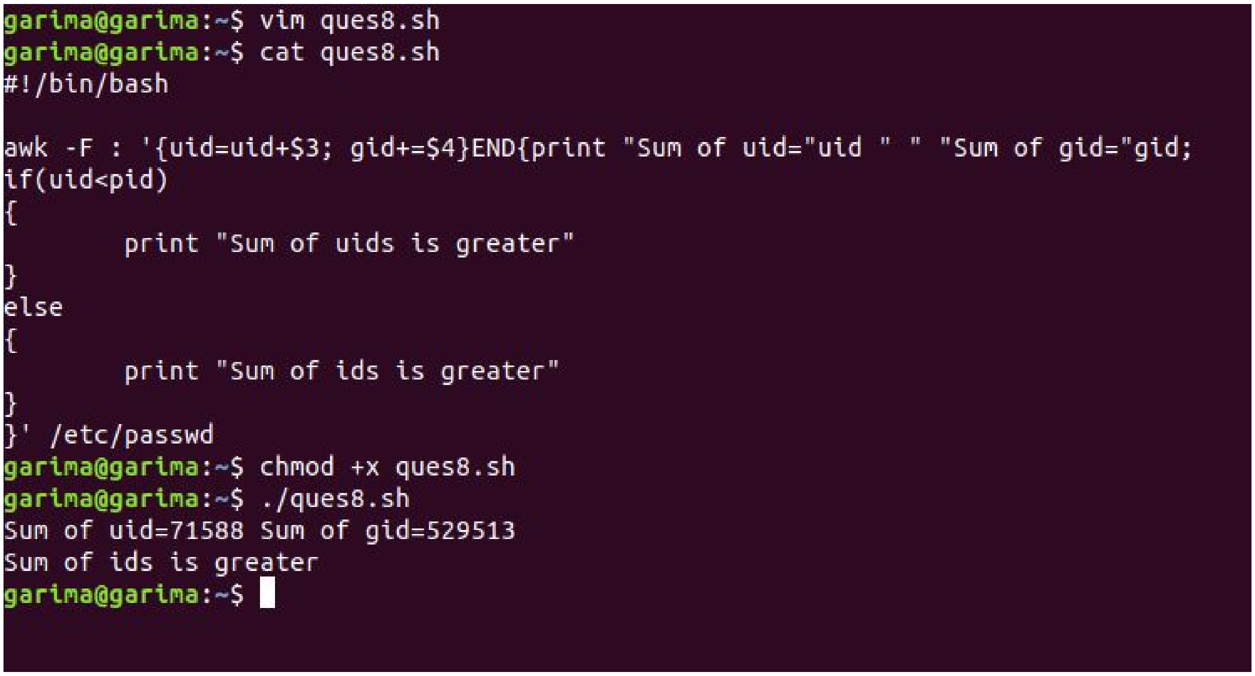
```
#!/bin/bash

while [ 1 ]
do
        echo "enter + for addition";
        echo "enter - for sub";
        echo "enter * for multi";
        echo "enter / for division";
        read a;
        case $a in
                +) echo "enter numbers";
                        read b;
                        read c;
                        echo `expr $b + $c`;
                        ;;
                -) echo "enter numbers";
                        read b;
                        read c;
                        echo `expr $b - $c`;
                        ;;
                \*) echo "enter numbers";
                        read b;
                        read c;
                        echo `expr $b \* $c`;
                        ;;
                /)  echo "enter numbers";
                        read b;
                        read c;
                        echo `expr $b / $c`;
                        ;;
                *) echo "wrong Input"
                        ;;
        esac
done
```

5. Write proper help documentation and print it with -h for the above script.



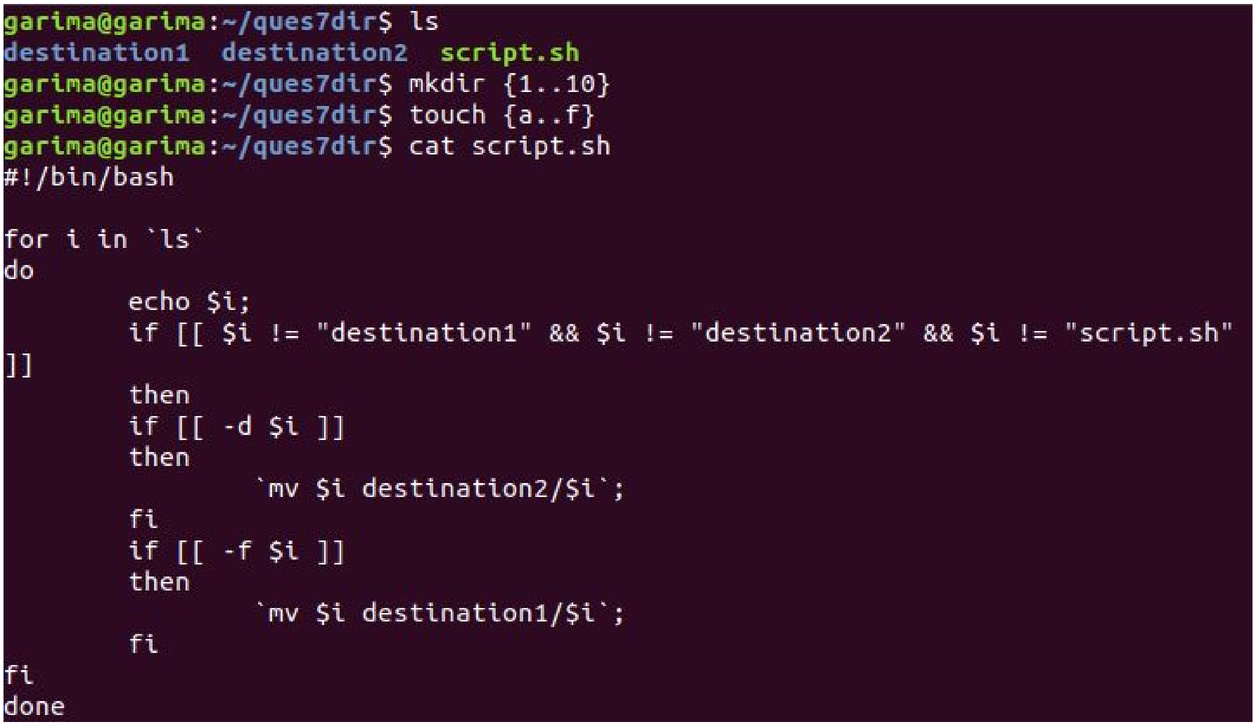
```
garima@garima:~$ vim ques7.sh
garima@garima:~$ chmod +x ques7.sh
garima@garima:~$ cat ques7.sh
#!/bin/bash
if [ "$1" == "-h" ]
then
        echo "this is the documentation for calculator"
fi
garima@garima:~$ ./ques7.sh -h
this is the documentation for calculator
garima@garima:~$ 
```

6. Create a script which takes input of "/etc/passwd" file and find out and print the sum of uids and gids. The script should tell which sum is greater.

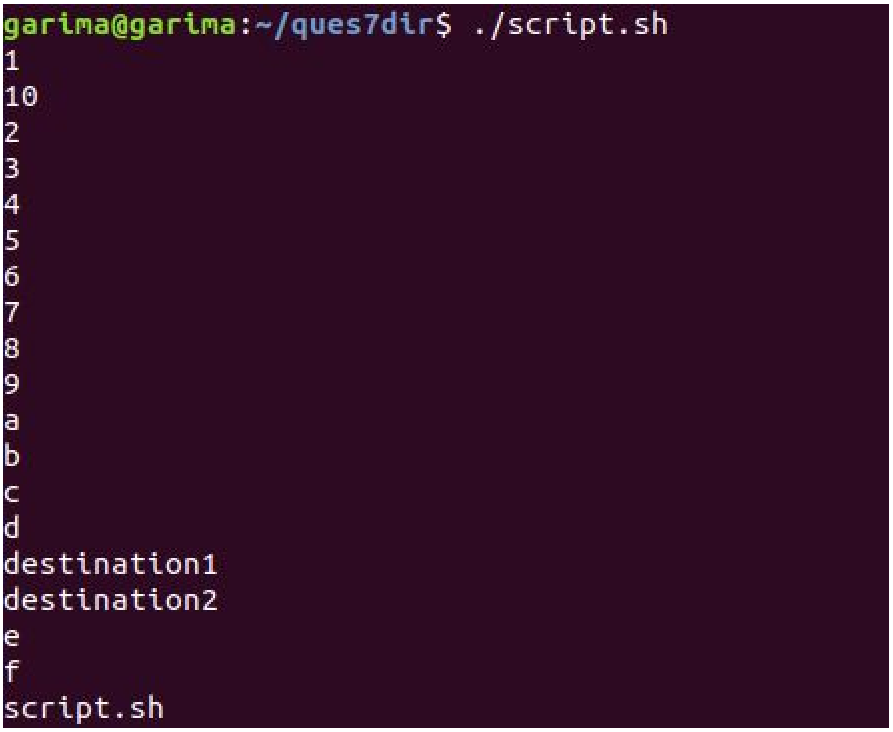


```
garima@garima:~$ vim ques8.sh
garima@garima:~$ cat ques8.sh
#!/bin/bash

awk -F : '{uid=uid+$3; gid+=$4}END{print "Sum of uid="uid " " "Sum of gid="gid;
if(uid<pid)
{
        print "Sum of uids is greater"
}
else
{
        print "Sum of ids is greater"
}
}' /etc/passwd
garima@garima:~$ chmod +x ques8.sh
garima@garima:~$ ./ques8.sh
Sum of uid=71588 Sum of gid=529513
Sum of ids is greater
garima@garima:~$ 
```

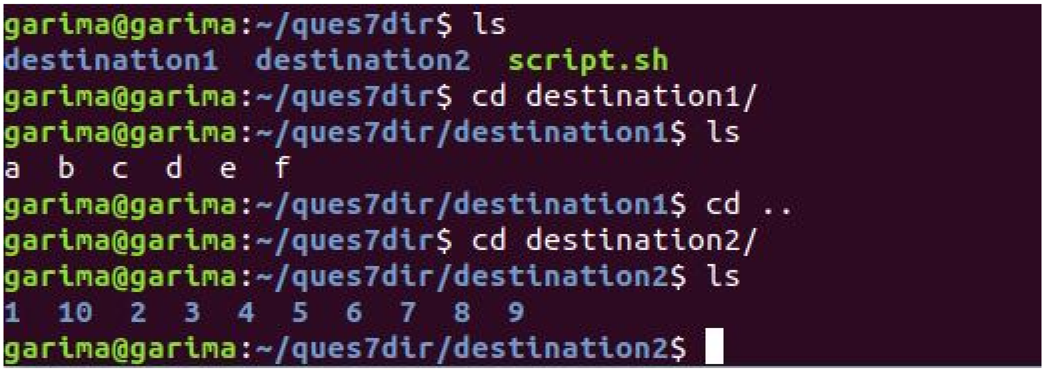
7. A directory contains files and sub-directories. Move files to destination1 and directories to destination2.



```
garima@garima:~/ques7dir$ ls
destination1  destination2  script.sh
garima@garima:~/ques7dir$ mkdir {1..10}
garima@garima:~/ques7dir$ touch {a..f}
garima@garima:~/ques7dir$ cat script.sh
#!/bin/bash

for i in `ls`
do
        echo $i;
        if [[ $i != "destination1" && $i != "destination2" && $i != "script.sh"
]]
        then
        if [[ -d $i ]]
        then
                `mv $i destination2/$i`;
        fi
        if [[ -f $i ]]
        then
                `mv $i destination1/$i`;
        fi
fi
done
```

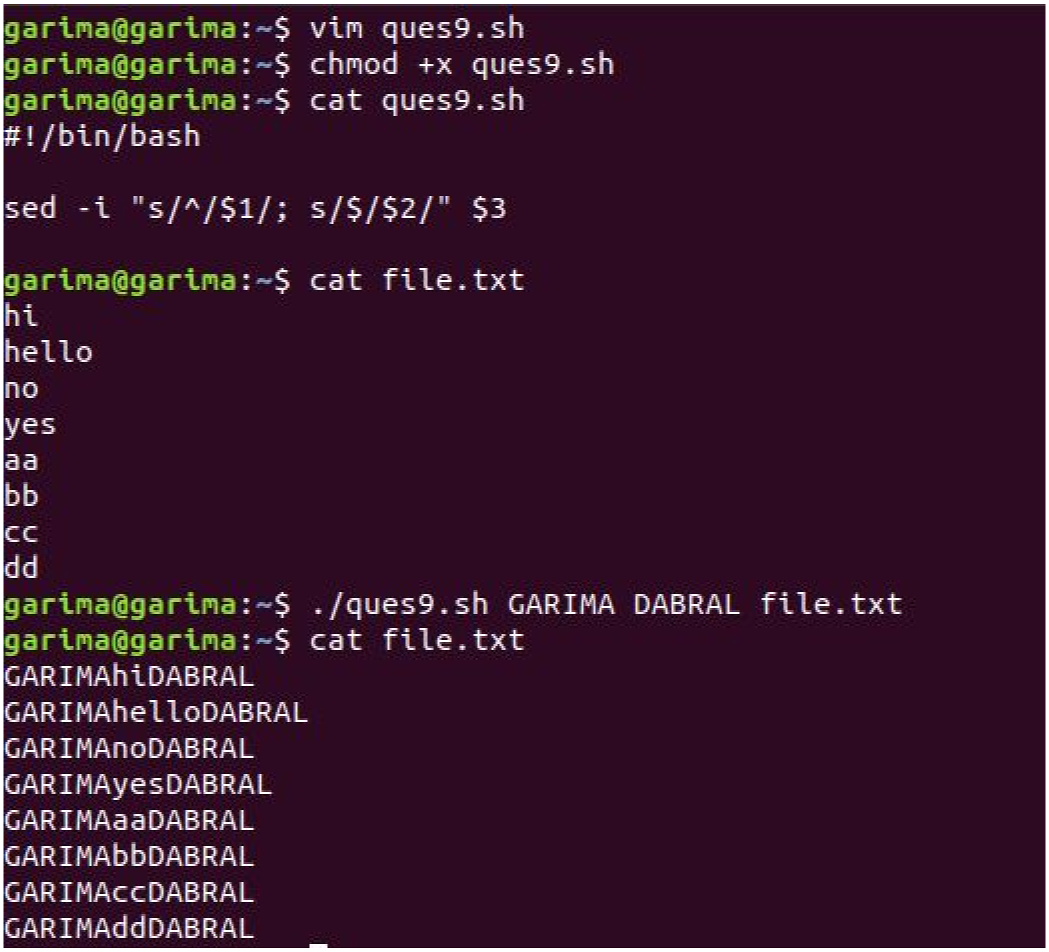


```
garima@garima:~/ques7dir$ ./script.sh
1
10
2
3
4
5
6
7
8
9
a
b
c
d
destination1
destination2
e
f
script.sh
```



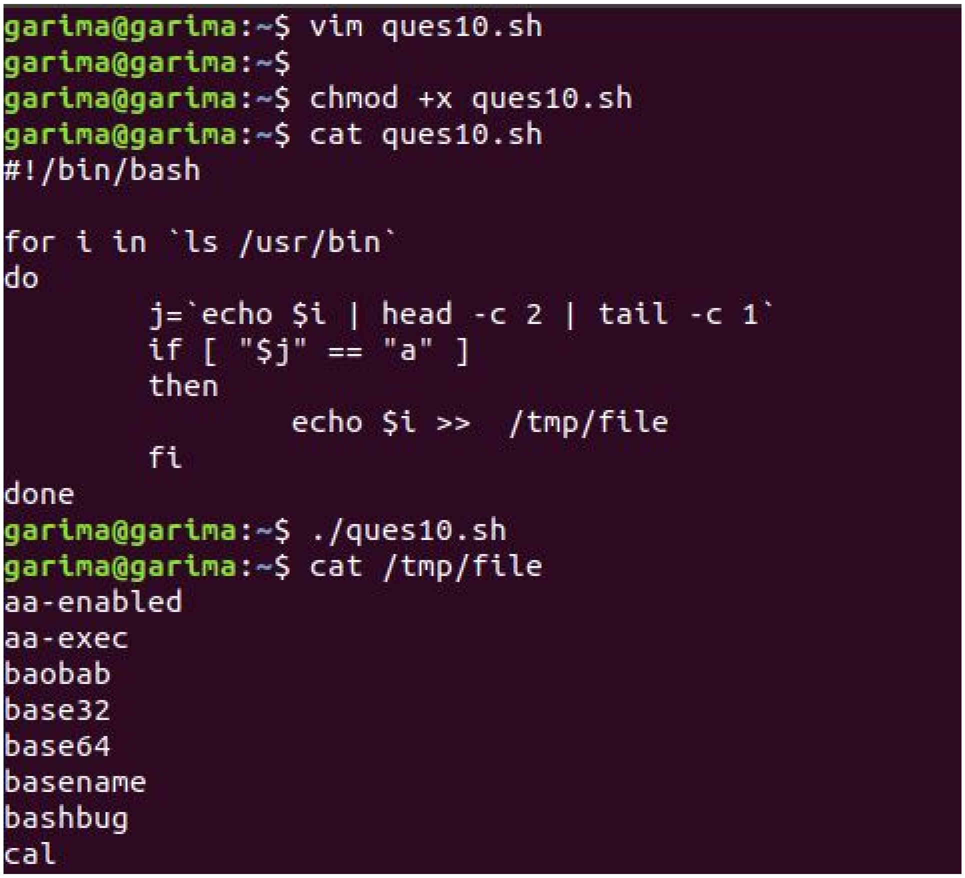
```
garima@garima:~/ques7dir$ ls
destination1  destination2  script.sh
garima@garima:~/ques7dir$ cd destination1/
garima@garima:~/ques7dir/destination1$ ls
a  b  c  d  e  f
garima@garima:~/ques7dir/destination1$ cd ..
garima@garima:~/ques7dir$ cd destination2/
garima@garima:~/ques7dir/destination2$ ls
1  10  2  3  4  5  6  7  8  9
garima@garima:~/ques7dir/destination2$
```

8. Create a script which takes three arguments, append first argument to every line in a file and second argument to the end of every line of the same file..



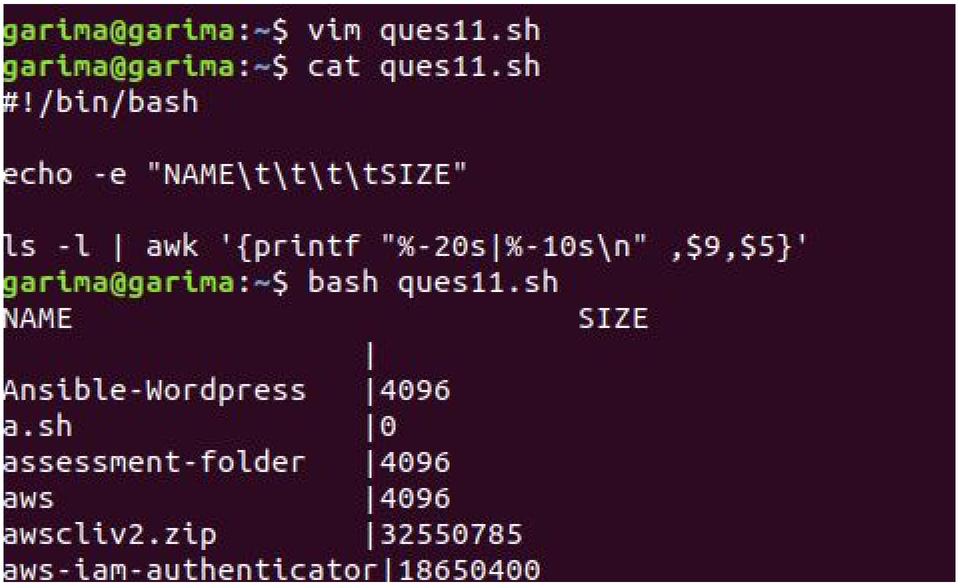
```
garima@garima:~$ vim ques9.sh
garima@garima:~$ chmod +x ques9.sh
garima@garima:~$ cat ques9.sh
#!/bin/bash

sed -i "s/^/$1/; s/$/$2/" $3

garima@garima:~$ cat file.txt
hi
hello
no
yes
aa
bb
cc
dd
garima@garima:~$ ./ques9.sh GARIMA DABRAL file.txt
garima@garima:~$ cat file.txt
GARIMAhiDABRAL
GARIMAhelloDABRAL
GARIMAnoDABRAL
GARIMAyesDABRAL
GARIMAaaDABRAL
GARIMAbbDABRAL
GARIMAccDABRAL
GARIMAddDABRAL
```

9. Make a list of files in /usr/bin that have the letter "a" as the second character. Put the result in a temporary file.



```
garima@garima:~$ vim ques10.sh
garima@garima:~$
garima@garima:~$ chmod +x ques10.sh
garima@garima:~$ cat ques10.sh
#!/bin/bash

for i in `ls /usr/bin`
do
        j=`echo $i | head -c 2 | tail -c 1`
        if [ "$j" == "a" ]
        then
                echo $i >>  /tmp/file
        fi
done
garima@garima:~$ ./ques10.sh
garima@garima:~$ cat /tmp/file
aa-enabled
aa-exec
baobab
base32
base64
basename
bashbug
cal
```

10. List all files in your home directory and print name and size in a table format.



```
garima@garima:~$ vim ques11.sh
garima@garima:~$ cat ques11.sh
#!/bin/bash

echo -e "NAME\t\t\t\tSIZE"

ls -l | awk '{printf "%-20s|%-10s\n" ,$9,$5}'
garima@garima:~$ bash ques11.sh
NAME                            SIZE
                    |
Ansible-Wordpress   |4096
a.sh                |0
assessment-folder   |4096
aws                 |4096
awscliv2.zip        |32550785
aws-iam-authenticator|18650400
```