ASSESSMENT ON: SHELL SCRIPTING

1. (output to terminal)Write a script to print:
a. “Welcome to Intelligrape”
b. <username>@<hostname>:<your present working directory>

```
#!/bin/bash
echo "Welcome to Intelligraph"
user=$(whoami)
host=$(hostname)
pswd=$(pwd)
echo $user"@"$host":"$pswd
~
~
~
~
~
"f.sh" 6L, 114C written
```

```
diksha@diksha:~$ ./f.sh
Welcome to Intelligraph
diksha@diksha:/home/diksha
diksha@diksha:~$
```

2 (arguments)Write a script

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

```
#!/bin/bash
echo "First argument=$1"
echo "Second argument=$2"
~
~
~
~
~
~
~
~
"f2.sh" [New] 3L, 63C written
```

```
diksha@diksha:~$ ./f2.sh 2 5
First argument=2
Second argument=5
diksha@diksha:~$
```

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

```
#!/bin/bash
if [ $# -gt 2 ]
then
        echo "Error"
        echo "No. of arguments $#"
        exit 2
else
        echo "Success"
        echo "Arguments are $1 $2"
fi
~
"f3.sh" [New] 10L, 135C written
```

```
diksha@diksha:~$ ./f3.sh 2 5
Success
Arguments are 2 5
diksha@diksha:~$ ./f3.sh 2 5 6 7
Error
No. of arguments 4
diksha@diksha:~$
```

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.

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

```
#!/bin/bash
sum()
{
        x=$1
        y=$2
        sum=`expr $x + $y`
        echo "$sum"
}

if [[ "$1" =~ ^[0-9]+$ ]] && [[ "$2" =~ ^[0-9]+$ ]]
then
        sum $1 $2
else
        echo "Both are not integer"
        echo "Not an integer:- $1,$2" >> error.log
fi
~
~
```

```
diksha@diksha:~$ ./f2.sh 5 f
Both are not integer
diksha@diksha:~$ cat e
error.log         errors.txt        err.txt           examples.desktop
diksha@diksha:~$ cat error.log
Not an integer:- 5,20
Not an integer:- 5,f
Not an integer:- 5,f
Not an integer:- 5,4
Not an integer:- 5,f
Not an integer:- 5,f
diksha@diksha:~$ ./f2.sh 5 7
12
diksha@diksha:~$
```

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

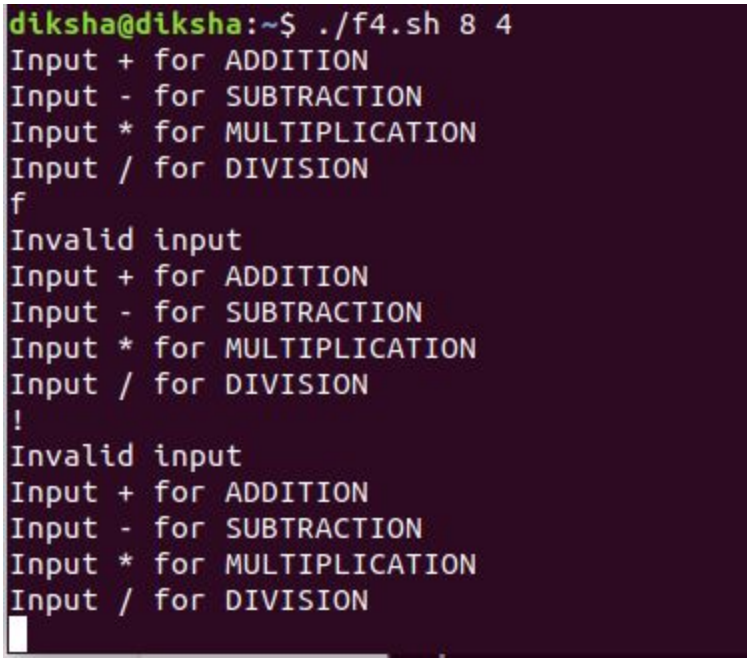
a. the script should ask user which operation the user wants to perform:+,-,*,/

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

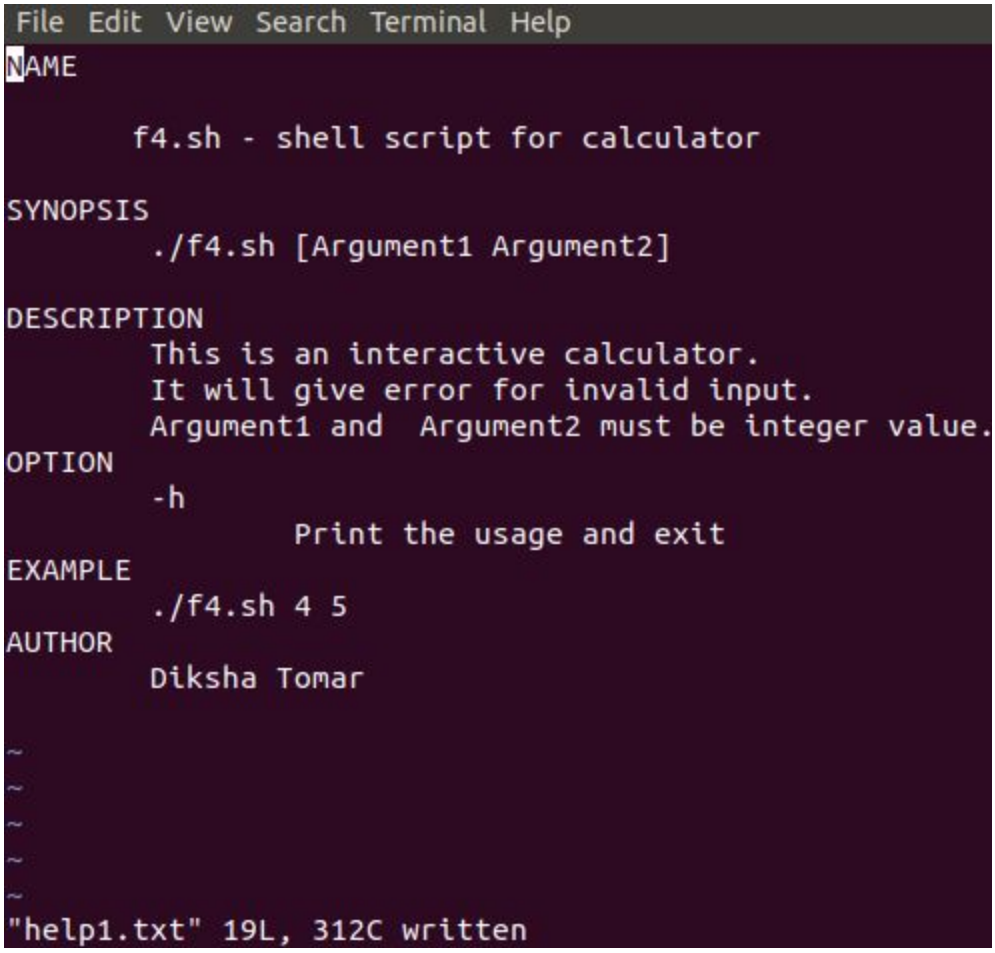
c. Use case statement instead of if.

```
#!/bin/bash
while [ 1 ]
do
        echo "Input + for ADDITION";
        echo "Input - for SUBTRACTION";
        echo "Input * for MULTIPLICATION";
        echo "Input / for DIVISION";
        read ch
        case "${ch}" in
                "+") echo "Result is" `expr $1 + $2`
                        exit 0;;
                "-") echo "Result is" `expr $1 - $2`
                        exit 0;;
                "*") echo "Result is" `expr $1 \* $2`
                        exit 0;;
                "/") echo "Result is" `expr $1 / $2`
                        exit 0;;
                *) echo "Invalid input";
        esac
done
```

```
diksha@diksha:~$ ./f4.sh 8 4
Input + for ADDITION
Input - for SUBTRACTION
Input * for MULTIPLICATION
Input / for DIVISION
*
Result is 32
diksha@diksha:~$ ./f4.sh 8 4
Input + for ADDITION
Input - for SUBTRACTION
Input * for MULTIPLICATION
Input / for DIVISION
+
Result is 12
diksha@diksha:~$ ./f4.sh 8 4
Input + for ADDITION
Input - for SUBTRACTION
Input * for MULTIPLICATION
Input / for DIVISION
-
Result is 4
diksha@diksha:~$ ./f4.sh 8 4
Input + for ADDITION
Input - for SUBTRACTION
Input * for MULTIPLICATION
Input / for DIVISION
/
Result is 2
```



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


File Edit View Search Terminal Help
NAME
f4.sh - shell script for calculator
SYNOPSIS
./f4.sh [Argument1 Argument2]
DESCRIPTION
This is an interactive calculator.
It will give error for invalid input.
Argument1 and Argument2 must be integer value.
OPTION
-h
Print the usage and exit
EXAMPLE
./f4.sh 4 5
AUTHOR
Diksha Tomar
~
~
~
~
~
"help1.txt" 19L, 312C written

```
#!/bin/bash
if [ "${1}" == "-h" ]
then
        less help1.txt
else
while [ 1 ]
do
        echo "Input + for ADDITION";
        echo "Input - for SUBTRACTION";
        echo "Input * for MULTIPLICATION";
        echo "Input / for DIVISION";
        read ch
        case "${ch}" in
                "+") echo "Result is" `expr $1 + $2`
                        exit 0;;
                "-") echo "Result is" `expr $1 - $2`
                        exit 0;;
                "*") echo "Result is" `expr $1 \* $2`
                        exit 0;;
                "/") echo "Result is" `expr $1 / $2`
                        exit 0;;
                *) echo "Invalid input";
        esac
done
fi

~
~
~
~
"f4.sh" 26L, 477C written
```

```
diksha@diksha:~$ ./f4.sh -h
```

```
NAME

       f4.sh - shell script for calculator

SYNOPSIS
        ./f4.sh [Argument1 Argument2]

DESCRIPTION
        This is an interactive calculator.
        It will give error for invalid input.
        Argument1 and  Argument2 must be integer value.
OPTION
        -h
                Print the usage and exit
EXAMPLE
        ./f4.sh 4 5
AUTHOR
        Diksha Tomar

 ESC
```

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 of greater.

```
#!/bin/bash
awk -F: '{uid+=$3; pid+=$4}END{print "uids="uid" " "pids="pid" ";
if(uid>pid){
        print uid
}
else{
        print pid
}
}' /etc/passwd
```

```
diksha@diksha:~$ ./f5.sh
uids=71587 pids=464106
464106
```

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

```
diksha@diksha:~/sort$ ls
a  aa  b  bb  c  cc  d  dd  destination1  destination2  e  ee  f6.sh  ff
```

```
#!/bin/bash
for i in `ls`
do
        if [[ "$i" != "destination1" && "$i" != "destination2" && "$i" != "f6.sh" ]]
        then
                if [ -f $i ]
                then
                        mv $i destination1/$i
                fi
                if [ -d $i ]
                then
                        mv $i destination2/$i
                fi
        fi
done
~
~
~
"f6.sh" 15L, 226C                                             8,21-42       All
```

```
diksha@diksha:~/sort$ ./f6.sh
diksha@diksha:~/sort$ ls
destination1  destination2  f6.sh
diksha@diksha:~/sort$
```

```
diksha@diksha:~/sort/destination1$ ls
a  b  c  d  e
diksha@diksha:~/sort/destination1$ cd ../destination2
diksha@diksha:~/sort/destination2$ ls
aa  bb  cc  dd  ee  ff
diksha@diksha:~/sort/destination2$
```

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

```
#!/bin/bash
sed -i "s/^/$1/; s/$/$2/" $3
~
```

```
diksha@diksha:~/sort$ cat >file.txt
hello
hi
watsup^C
diksha@diksha:~/sort$ chmod +x f7.sh
diksha@diksha:~/sort$ ./f7.sh hi diksha file.txt
diksha@diksha:~/sort$ cat file.txt
hihellodiksha
hihidiksha
diksha@diksha:~/sort$
```

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.

```
for i in `ls /usr/bin`
do
        j=`echo $i | head -c 2 | tail -c 1`
        if [ "$j" == "a" ]
                then
                        echo $i >> /tmp/abc
        fi
done
~
~
```

```
diksha@diksha:/tmp$ cat abc
aa-enabled
aa-exec
baobab
base32
base64
basename
bashbug
cal
calendar
calibrate_ppa
canberra-gtk-play
cancel
captoinfo
catchsegv
catman
cautious-launcher
factor
faillog
```

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

```
echo -e "Name\t\t\t\tSize"
ls -l | awk 'printf "%-30s|%-18s\n" ,$9,$5'
~
~
```

```
diksha@diksha:~$ vim f9.sh
diksha@diksha:~$ ./f9.sh
Name                                Size
                                   |
abcd                               |10
ABCD.txt                           |20
abc.sh                             |4
a.txt                              |6
awscliv2.zip                       |32550785
aws-iam-authenticator              |18650400
ball.tar.gz                        |45
b.txt                              |3
cat                                |9
destination1                       |4096
destination2                       |4096
error.log                          |0
errors.txt                         |0
err.txt                            |57
examples.desktop                   |8980
```