

NAME : ABUTHAHIR M
BITS ID : 2019AT15025

SET - 1

1. Write a menu driven shell script which accepts a number N and an option to calculate the average of:

(a) first 'N' odd numbers

(b) first 'N' even numbers

Hint: Solve the above question using loops

Shell Script:

```
echo "Enter the Size of Input Numbers"
read N
echo "Press 1 to calculate average of odd numbers"
echo "Press 2 to calculate average of odd numbers"
read option
i=1
sum=0

echo "Enter $N numbers"

if [ $option -eq 1 ]
then
while [ $i -le $N ]
do
read number
if [ $i%2 == 1 ]
then
sum=$((sum+number))
fi

i=$((i + 1))
done
```

```
average=$sum/$i

echo "Average of Odd numbers is $average"
elif[ $option -eq 2 ]

while [ $i -le N ]
do
read number
if[ $i%2 == 0]

sum=$((sum+number))
i=$((i + 1))

done
fi
average=$sum/$N

echo "Average of Even numbers is $average"

fi
```

2. Following is the first four records in a file “data.txt”.

71723 Ram Sen 70 72 75

91924 Raghubir Yadav 82 73 80

53425 Ram Chauhan 93 81 86

44917 Ratan Yadav 95 79 91

Each record contains ID (5 chars), 1 space, First name (10 chars),1 space, Second name (8 chars),1space, marks in physics (3 chars),

1space, marks in chemistry (3 chars), 1space, marks in mathematics(3 chars) and a newline character.

Write a shell program info.sh to achieve the following

- i) If the program is run without any argument (sh info.sh), it will display the first name, second name and average score of each student in the file.**
- ii) If the program is run with numeric argument (sh info.sh 44917), it will assume it as ID of student and output ID, first name, second name and the average score of that student. If ID does not match, the program should display “record not found”.**
- iii) If the program is run with non-numeric argument (sh info.sh yadav), it will assume it as either first or second name and output the ID’s of all the students whose first or last name matches with the argument (irrespective of case)**
- iv) If the number of arguments is greater than one, the program should display an error message.**

3. Write a shell script to

(a) count the number of .doc, .txt, .c files in current working directory

```
ls -lR /path/to/dir/*.doc | wc -l  
ls -lR /path/to/dir/*.txt | wc -l  
ls -lR /path/to/dir/*.c | wc -l
```

(b) convert all the files with extension .c to .doc.

```
#!/bin/bash  
for f in *.c;  
do  
    mv -- "$f" "$(basename -- "$f" .c).doc"  
done
```

4. Consider a database which contains following fields: Roll number, name, semester and marks

in mathematics. Write a menu driven shell script for

(a) adding a record

(b) deletion of a record corresponding to a roll number

(c) finding a record in a database corresponding to a roll number or ID Number

(d) exit

The shell script should accept a choice from the user and execute the corresponding option until exit option is chosen.

```
clear
i="y"
echo "Enter name of database "
read db
while [ $i = "y" ]
do
clear
echo "1.Add Records "
echo "2.deletion of a record corresponding to a roll number "
echo "3.finding a record in a database corresponding to a roll number or ID
Number "
echo "4.Exit "
echo "Enter your choice "
read ch
case $ch in
1)echo "Enter new std id "
iread tid
echo "Enter new name:"
read tnm
echo "Enter semester "
read sem
echo "Enter mathematics mark"
read math
echo "$tid $tnm $sem $math">>$db;;
2)echo "Enter Id"
read id
```

```

# set -a
# sed '/$id/d' $db>dbs1
grep -v "$id" $db >dbs1
echo "Record is deleted"
cat dbs1;;
3)echo "Enter id "
read id
grep -i "$id" $db;;
4)exit;;
*)echo "Invalid choice ";;
esac
echo "Do u want to continue ?"
read i
if [ $i != "y" ]
then
    exit
fi
done

```

5. Write a shell script to accept 2 matrices [2x2] and perform the following based on the menu option given.

- (a) addition**
- (b) subtraction**
- (c) multiplication**
- (d) transpose**
- (e) exit**

```
i="y"
```

```

for ((i = 0; i < 2; i++)); do
    for ((j = 0; j < 2; j++)); do
        read a[$((i * 2 + j))]
    done
done

```

```
done
done
for ((i = 0; i < 2; i++)); do
    for ((j = 0; j < 2; j++)); do
        read b[$((i * 2 + j))]
    done
done
```

```
while [ $i = "y" ]
```

```
do
```

```
echo "1.Addition"
```

```
echo "2.Subtraction"
```

```
echo "3.Multiplication"
```

```
echo "4.Transpose"
```

```
echo "5.Exit"
```

```
echo "Enter your choice"
```

```
read ch
```

```
case $ch in
```

```
    1)k=0
    c=()
    for((i=0; i<rows; i++))
    do
```

```

for((j=0; j<cols; j++))
do
    index=$((i*cols+j))
    c[k]=$(( ${a[index]} + ${b[index]} ))
    k=$((k+1))
done
done
echo "Addition of two matrix"
for((i=0; i<rows; i++))
do
    for((j=0; j<cols; j++))
    do
        index=$((i*cols+j))
        echo -n "${c[index]} "
    done
    echo
done

```

```

    2)k=0
c=()
for((i=0; i<rows; i++))
do
    for((j=0; j<cols; j++))
    do
        index=$((i*cols+j))
        c[k]=$(( ${a[index]} - ${b[index]} ))
        k=$((k+1))
    done
done
echo "Subtraction of two matrix"
for((i=0; i<rows; i++))
do
    for((j=0; j<cols; j++))

```



```

do
    index=$((i*cols+j))
    echo -n "${c[index]} "
done
echo
done

```

```

3)k=0
c=()
for((i=0; i<rows; i++))
do
    for((j=0; j<cols; j++))
    do
        index=$((i*cols+j))
        c[k]=$(( ${a[index]} * ${b[index]} ))
        k=$((k+1))
    done
done
echo "Multiplication of two matrix"
for((i=0; i<rows; i++))
do
    for((j=0; j<cols; j++))
    do
        index=$((i*cols+j))
        echo -n "${c[index]} "
    done
done
echo
done

```

```

4)i=0 ; k=0
echo "Transpose of a Matrix"
until [ $i 'eq' $c ]
do

```

```

j=0;
until [ $j 'eq $r ]
do
    n= <code>expr $j \* $c
    m= `expr $n + $i
    b[$k]=${a[$m]}
    echo "${b[$k]} \t"
    k=<code>expr $k + 1
    j=<code>expr $j + 1
done
i=<code>expr $i + 1
echo "\n"
done
5)exit

*)echo "Invalid choice";;

esac

echo "Do u want to continue ?[y/n]"

read i

if [ $i != "y" ]

then

    exit

fi

done

```

