

WEEK 3**Lab Exercises:**

Write shell scripts to perform the following:

1: Find whether the given number is even or odd.

```
echo "Enter the number to check whether its even or odd: "  
read dan  
temp=$(( $dan%2 ))  
if [ $temp -eq 1 ]  
then  
echo "The number entered is ODD"  
else  
echo "The number entered is EVEN"  
fi
```

```
Student@dblab-hp-28:~/190905513$ ./mde1.sh  
Enter the number to check whether its even or odd:  
5  
The number entered is ODD  
Student@dblab-hp-28:~/190905513$ ./mde1.sh  
Enter the number to check whether its even or odd:  
8  
The number entered is EVEN  
Student@dblab-hp-28:~/190905513$
```

2: Print the first 'n' odd numbers.

```
echo "Enter the number of odd numbers you want to print: "  
read num  
x=0  
odd=0  
echo "Printing $num ODD numbers"  
while [ $x -lt $num ]  
do  
odd=`expr $x \* 2 + 1`  
echo $odd  
x=$(( x+1 ))  
done
```

```
Student@dblab-hp-28:~/190905513$ nano mde2.sh  
Student@dblab-hp-28:~/190905513$ ./mde2.sh  
Enter the number of odd numbers you want to print:  
5  
Printing 5 ODD numbers  
1  
3  
5  
7  
9  
Student@dblab-hp-28:~/190905513$
```

3: Find all the possible quadratic equation roots using case.

```
echo "Enter Coefficients of Quad Eqn namely a, b and c:"
read a
read b
read c
e=$(( b * b - 4 * a * c ) | bc
echo $e
#d=`expr $b * $b - 4 * $a * $c`
if[ $e -ge 0 ]
then qd="r"
else qd="i"
fi

case $quad in

"r")
sq=`echo "sqrt($d)" | bc`
r1=`echo "(-1*$b+$sq)/(2*$a)" | bc`
r2=`echo "(-1*$b-$sq)/(2*$a)" | bc`

echo "Root 1: $r1 Root 2: $r2";;

"i")
de=`expr -1 * $d`
sq=`echo "sqrt($de)" | bc`
re=`echo "(-1*$b)/(2*$a)" | bc`
im=`echo "$sq/(2*$a)" | bc`
echo "Root 1: $re + i$im Root 2: $re -i$im";;
esac
```

//Error while execution

4: Find the factorial of a given number.

```
echo "Enter the number to find its factorial: "  
read num  
i=0  
fact=1  
while [ $i -lt $num ]  
do  
i=$((i+1)) fact=`expr $i \* $fact`  
done  
echo "The factorial of $num is equal to $fact"
```

```
Student@dblab-hp-28:~/190905513$ ./mde4.sh  
Enter the number to find its factorial:  
5  
The factorial of 5 is equal to 120  
Student@dblab-hp-28:~/190905513$ ./mde4.sh  
Enter the number to find its factorial:  
3  
The factorial of 3 is equal to 6  
Student@dblab-hp-28:~/190905513$
```

WEEK 4**Experiment:**

1: Write a program in c, to implement a stack with push and pop operations using suitable functions. Create static libraries to perform various operations on the stack. Create a header file for function declaration.

//push.c

```
student@prg28:~$ cat push.c
#include<stdio.h>
void push(int a[],int *top,int n){
a[++(*top)]=n;
}
```

//pop.c

```
student@prg28:~$ cat pop.c
#include <stdio.h>
void pop(int a[],int *top){
printf("Popped element is %d\n",a[*top]);
(*top)--;
}
```

//display.c

```
student@prg28:~$ cat display.c
#include <stdio.h>
void display(int a[],int top){
int i;
printf("Stack:");
for(i=0;i<=top;i++){

printf("%d    ",a[i
]);
}
}
```

//stack.c

```
student@prg28:~$ cat stack.c
#include <stdio.h>
#include <stdlib.h>
#include "mystklib.h"
int main(){
int a[5];
int top=-1;
push(a,&top,5);
push(a,&top,2);
push(a,&top,7);
pop(a,&top);
display(a,top);
}
```

//library file i.e mystklib.h

```
student@prg28:~$ cat mystklib.h
void pop(int *, int *);
void push(int *, int *, int );
void display(int *,int );
student@prg28:~$
```

//output after running stack.c

```
student@prg28:~$ ./stack
Popped element is 7
Stack:5    2    student@prg28:~$
```

//Makefile1

```
student@prg28:~$ cat Makefile1
stack: stack.o pop.p push.o display.o mystklib.h
      gcc -o stack stack.o pop.o push.p display.o
stack.o: stack.c
      gcc -c stack.c
pop.o: pop.c
      gcc -c pop.c
push.o: push.c
      gcc -c push.c
display.o: display.c
      gcc -c display.o
mystklib: pop.o push.o display.o
      gcc -o mystklib.h pop.o push.o display.o
```