Name: Pravin NIrmal
PRN : 230341220109
Roll No :233175

Assignments : OS (Operating System)

A. Create a basic calculator with using case.

```
#!/bin/bash
read -p  "Enter a two number: " a b
echo "1.addition"
echo "2.Subtraction"
echo "3.division"
echo "4.Multiply"
echo "Choice : "
read ch
case $ch in
1)
     expr $a + $b
    ;;
2)
    expr $a - $b
    ;;
3)
    expr $a / $b
    ;;
4)
    expr $a \* $b
    ;;
*)
    echo "Invalid choice "
Esac
```

Output:

```
Enter a two number: 5 2
1.addition
2.Subtraction
3.division
4.Multiply
Choice :
1
7
```

B. Find out the greatest number among three numbers entered by users using if condition

```
#!/bin/bash
echo "Enter three numbers to find greast: "
read a b c
if (( a >= b && a >= c ))
then
   echo "$a is greatest number"
elif (( b >= a && b >= c ))
then
   echo "$b is greatest"
else
   echo "$c is greatest"
fi
```

Output :

```
Enter three numbers to find greast:
1 2 3
3 is greatest
```

C. Write a program to take input of number from user and generate that number of .txt files.

```
#!/bin/bash

read -p "Enter 5 numbers to enter in text file: " a b c d e
touch data.txt
echo "$a" > data.txt
echo "$b" >> data.txt
echo "$c" >> data.txt
echo "$d" >> data.txt
echo "$e" >> data.txt
cat data.txt
```

D. Write a program to check whether the number is even or odd?

```
#!/bin/bash
read -p "Enter a number: " i
   if (( $i % 2 == 0 ))
   then
   echo "$i even number"
```

```
    else
        echo "$i odd number "
    fi
```

Output :

```
Enter a number: 5
5 odd number
```

---

1. Write a Shell Script to display the first 10 natural numbers.

```
#! /bin/bash
for (( i=1; i<11; i++ ))
do
    echo “$i”
done
```

Output :

```
1 2 3 4 5 6 7 8 9 10
```

---

2. Write a Shell Script to compute the sum of the first 10 natural numbers.

```
#! /bin/bash
sum=0
for (( i=0; i<11; i++ ))
do
    sum=`expr $sum + $i`
done
echo "$sum"
```

Output :

```
55
```

3. Write a Shell Script to display n terms of natural numbers and their sum.
Test Data : 7
Expected Output :
The first 7 natural number is :
1 2 3 4 5 6 7
The Sum of Natural Number upto 7 terms : 28

```
#!/bin/bash
./sumnatural

read -p  "enter a numbers "  b
for (( i=1; i<=$b; i++ ))
do
   sum=$((sum+i))
done
```

Outpput :

```
enter a numbers 5
15
```

---

4. Write a Shell Script to read 10 numbers from the keyboard and find their sum and average.

```
#!/bin/bash
echo "How many numbers you want to add"
read num
sum=0
for (( i=0; i<num; i++ ))
do
   read a
   sum=`expr $sum + $a`
done
echo "$sum"
```

Output :

```
How many numbers you want to add
5
1
```

2
3
4
6
16

---

5. Write a Shell Script to display the cube of the number up to an integer.
Test Data :
Input number of terms : 5
Expected Output :
Number is : 1 and cube of the 1 is :1
Number is : 2 and cube of the 2 is :8
Number is : 3 and cube of the 3 is :27
Number is : 4 and cube of the 4 is :64
Number is : 5 and cube of the 5 is :125

```
#!/bin/bash
echo "Cubes  : "
for (( i=1; i<6; i++ ))
do
   echo -e  "Number is : $i and the cube of $i is : `expr $i \* $i \* $i`"
done
```

Output :

```
Cubes  :
Number is : 1 and the cube of 1 is : 1
Number is : 2 and the cube of 2 is : 8
Number is : 3 and the cube of 3 is : 27
Number is : 4 and the cube of 4 is : 64
Number is : 5 and the cube of 5 is : 125
```

---

6. Write a Shell Script to display the multiplication table for a given integer.
Test Data :
Input the number (Table to be calculated) : 15
Expected Output :
15 X 1 = 15
...
...
15 X 10 = 150

```
read -p "Enter two numbers " i
for (( j=1; j<11; j++ ))
   do
         echo -n " $i * $j = `expr $i \* $j` "
         echo  -n " ; "
   done
```

Output :

```
 2 * 1 = 2  ;  2 * 2 = 4  ;  2 * 3 = 6  ;  2 * 4 = 8  ;  2 * 5 = 10  ;  2 * 6 = 12  ;  2 * 7 = 14  ;  2 * 8 = 16
;  2 * 9 = 18  ;  2 * 10 = 20
```

---

7. Write a Shell Script to display the multiplier table vertically from 1 to n.
Test Data :
Input upto the table number starting from 1 : 8
Expected Output :
Multiplication table from 1 to 8
1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8
...
1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80

```
#!/bin/bash
read -p "Enter two numbers " a b
for (( i=a; i<=b; i++ ))
do
   for (( j=1; j<11; j++ ))
   do
         echo -n " $i * $j = `expr $i \* $j` "
         echo  -n " ; "
   done
done
```

Output :

```
Enter two numbers 1 5
 1 * 1 = 1  ;  1 * 2 = 2  ;  1 * 3 = 3  ;  1 * 4 = 4  ;  1 * 5 = 5  ;  1 * 6 = 6  ;  1 * 7 = 7  ;  1 * 8 = 8  ;  1
* 9 = 9  ;  1 * 10 = 10  ;  2 * 1 = 2  ;  2 * 2 = 4  ;  2 * 3 = 6  ;  2 * 4 = 8  ;  2 * 5 = 10  ;  2 * 6 = 12  ;
2 * 7 = 14  ;  2 * 8 = 16  ;  2 * 9 = 18  ;  2 * 10 = 20  ;  3 * 1 = 3  ;  3 * 2 = 6  ;  3 * 3 = 9  ;  3 * 4 =
```

12 ; 3 * 5 = 15 ; 3 * 6 = 18 ; 3 * 7 = 21 ; 3 * 8 = 24 ; 3 * 9 = 27 ; 3 * 10 = 30 ; 4 * 1 = 4 ; 4 * 2 = 8 ; 4 * 3 = 12 ; 4 * 4 = 16 ; 4 * 5 = 20 ; 4 * 6 = 24 ; 4 * 7 = 28 ; 4 * 8 = 32 ; 4 * 9 = 36 ; 4 * 10 = 40 ; 5 * 1 = 5 ; 5 * 2 = 10 ; 5 * 3 = 15 ; 5 * 4 = 20 ; 5 * 5 = 25 ; 5 * 6 = 30 ; 5 * 7 = 35 ; 5 * 8 = 40 ; 5 * 9 = 45 ; 5 * 10 = 50 ;

---

8. Write a Shell Script to display the n terms of odd natural numbers and their sum.
Test Data
Input number of terms : 10
Expected Output :
The odd numbers are :1 3 5 7 9 11 13 15 17 19
The Sum of odd Natural Number upto 10 terms : 100

```
echo " enter the 10 natural no "
 read a b c d e f g h i j
sum=`expr $a + $b + $c + $d + $e + $f + $g + $h + $i + $j`
 avg=`expr $sum / 10`
echo " sum = $sum Average = $avg "
```

Output:

```
 enter the 10 natural no
 1 2 3 4 5 6 7 8 9 10
 sum = 55
 Average = 5
```

---

9. Write a Shell Script to display a pattern like a right angle triangle using an asterisk.

The pattern like :

*****
******
****
****

```
#!/bin/bash
for (( i=0; i<6; i++ ))
do
   for (( j=0; j<$i; j++ ))
   do
          echo -n  " * "
   done
   echo -e "\n"
done
```

10. Write a Shell Script to display a pattern like a right angle triangle with a number.

The pattern like :

1
12
123
1234

```
#!/bin/bash
for (( i=1; i<6; i++ ))
do
   for (( j=1; j<=$i; j++ ))
   do
          echo -n "$j"
   done
   echo -e "\n"
done
```

---

11. Write a Shell Script to make such a pattern like a right angle triangle with a number which will repeat a number in a row.

The pattern like :

1
22
333
4444

```
#!/bin/bash
k=0;
for (( i=0; i<6; i++ ))
do
   k=`expr $k + 1`
   for (( j=0; j<=$i; j++ ))
   do
          echo -n "$k"
   done
   echo -e  "\n"
done
```

12. Write a Shell Script to make such a pattern like a right angle triangle with the number increased by 1.

The pattern like :

```
1
2 3
4 5 6
7 8 9 10
```

```
#!/bin/bash
k=1
for (( i=1; i<6; i++ ))
do
    for (( j=1; j<=i; j++ ))
    do
            echo -n "$k "
            k=`expr $k + 1`
    done
    echo -e "\n"
done
```

---

13. Write a Shell Script to make a pyramid pattern with numbers increased by 1.

```
  1
 2 3
4 5 6
7 8 9 10
```

```
#!/bin/bash
k=0
for (( i=1; i<7; i++ ))
do
    for (( j=7-i; j>0; j-- ))
    do
            echo -n "  "
    done
    for (( o=1; o<=i; o++ ))
    do
            k=`expr $k + 1`
            echo -n  " $k "
```

```
    done
    echo -e "\n"
done
```

---

14. Write a Shell Script to make such a pattern as a pyramid with an asterisk.

```
  *
 * *
* * *
* * * *
```

```
#!/bin/bash
for (( i=1; i<6; i++ ))
do
    for (( j=5-i; j>0; j-- ))
    do
            echo -n "  "
    done
    for (( k=1; k<=i; k++ ))
    do
            echo -n " *  "
    done
    echo -e "\n"
done
```

---

15. Write a Shell Script to calculate the factorial of a given number.
Test Data :
Input the number : 5
Expected Output :
The Factorial of 5 is: 120

```
#!/bin/bash
read -p " Enter a number " a
fact=1
for (( i=1; i<=$a; i++ ))
do
            fact=$(( fact * i ))
done
echo -e  "\n $fact"
```

Output:

Enter a number 5
120

---

16. Write a Shell Script to display the sum of n terms of even natural numbers.
Test Data :
Input number of terms : 5
Expected Output :
The even numbers are :2 4 6 8 10
The Sum of even Natural Number upto 5 terms : 30

```
#! /bin/bash
read -p "enter the no " n
sum=0
for (( i=2 ;i<=n*2;i++))
do
        if (( i%2 == 0))
        then
                echo -n " $i " sum=`expr $sum + $i`
        fi
done
echo
echo " sum = $sum "
```

Output:

enter the no 5
2 4 6 8 10
sum = 30

---

17. Write a Shell Script to make such a pattern like a pyramid with a number which will repeat the number in the same row.

1
2 2
3 3 3
4 4 4 4

#!/bin/bash

```
read -p " enter the n " n

for (( i=1 ;i<=n ;i++))

do

        for (( j=1;j<=n-i;j++))

        do

        echo -n " "

        done

          for (( k=1 ;k<=i;k++))

          do

                  echo -n " $i"

          done

          echo

done
```

---

18. Write a Shell Script to find the sum of the series [1-X^2/2!+X^4/4!- .....].
Test Data :
Input the Value of x :2
Input the number of terms : 5
Expected Output :
the sum = -0.415873
Number of terms = 5
value of x = 2.000000

```
#! /bin/bash

read -p " enter the value of x " n1

read -p " enter the no of terms " t1
```

```
sum=0
num=1
c=2
f=1
t=1
fact=1
for (( i=1;i<=t1;i++))
do

  for ((j=1;j<=c;j++))
  do
    fact=$((fact * j))
    num=$((num * n1))
  done
     f=$(( f * -1))
 num=$((num * f))
 t=$( echo "scale=2; $num / $fact " | bc)
#  echo $t
 sum=$( echo "scale=2; $sum + $t " | bc)
#   sum=$((sum + t))
            c=$((c+2))
            num=1
```

```
        fact=1

done

 sum=$( echo "scale=2; $sum + 1" | bc)

echo " sum = $sum "
```

Output:

```
 enter the value of x 5
 enter the no of terms 2
 sum = 14.54
```

---

19. Write a Shell Script to display the n terms of a harmonic series and their sum.
1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms
Test Data :
Input the number of terms : 5
Expected Output :
1/1 + 1/2 + 1/3 + 1/4 + 1/5 +
Sum of Series upto 5 terms : 2.283334

```
#! /bin/bash

sum=0

read -p " enter the no " n

for (( i=1;i<=n;i++))

do

        echo -n "1/$i + "

        t=$( echo " scale=6; 1 / $i" | bc )

        sum=$( echo "scale=6; $sum + $t " | bc )

done

echo
```

```
echo "Sum of Series upto $n terms = $sum "
```

Output :

```
 enter the no 5
1/1 + 1/2 + 1/3 + 1/4 + 1/5 +
Sum of Series upto 5 terms = 2.283333
```

---

20. Write a Shell Script to display the pattern as a pyramid using asterisks, with each row containing an odd number of asterisks.

```
  *
 ***
*****
```

```
#!/bin/bash
l=1
for (( i=1; i<=5; i++ ))
do
   for (( j=1; j<=5-i; j++ ))
   do
         echo -n " "
   done

   for (( k=1; k<=l; k++ ))
   do
         echo -n "*"
   done
   l=$(( l + 2 ))
   echo -e "\n"
done
```

---

21. Write a Shell Script to display the sum of the series [9 + 99 + 999 + 9999 ...].
Test Data :
Input the number or terms :5
Expected Output :
9 99 999 9999 99999
The sum of the saries = 111105

```
#!/bin/bash
```

```
read -p "Enter a number of terms : " num
l=1
echo  "Numbers are : "
for (( i<0; i<num; i++ ))
do
   for (( j=0, k=1; j<=i; j++ ))
   do
          k=$(( 10 * k ))
   done
   echo -n "$(( 9 * l ))  "
   l=$(( k + l ))
   sum=$(( sum + l ))
#    echo -e "\n"
done

echo "Sum  : $sum"
```

Output :

```
Enter a number of terms : 9
Numbers are :
9  99  999  9999  99999  999999  9999999  99999999  999999999  Sum  : 1234567899
```

---

22. Write a Shell Script to print Floyd's Triangle.

1
01
101
0101
10101

```
#!/bin/bash
for (( i=0; i<6; i++ ))
do

   for (( j=1; j<=$i; j++ ))
   do
          if (( $(( j % 2 )) == 0 ))
          then
                 echo -n  "0"
          else
                 echo -n  "1"
```

```
        fi
    done
    echo -e "\n"
Done
```

---

23. Write a Shell Script to find the sum of the series [x - x^3 + x^5 + ...].
Test Data :
Input the value of x :3
Input number of terms : 5
Expected Output :
The sum is : 16.375000

```
#! /bin/bash
read -p " enter the value of x " n1
read -p " enter the no of terms " t1
sum=0
num=1
c=2
f=1
t=1
fact=1
for (( i=1;i<=t1;i++))
do
        for ((j=1;j<=c;j++))
        do fact=$((fact * j))
                num=$((num * n1))
        Done
         f=$(( f * -1))
        num=$((num * f))
        t=$( echo "scale=2; $num / $fact " | bc)
        #echo $t
        sum=$( echo "scale=2; $sum + $t " | bc)
        # sum=$((sum + t)) c=$((c+2))
        num=1
        fact=1
done
sum=$( echo "scale=2; $sum + 1" | bc)
echo " sum = $sum "
```

Output:

```
eter the no of terms 5 sum = -.42
enter the value of x 2
```

enter the no of terms 2 sum = -.34

---

24. Write a Shell Script to find the sum of the series [x - x^3 + x^5 + ...].
Test Data :
Input the value of x :2
Input number of terms : 5
Expected Output :
The values of the series:
2
-8
32
-128
512
The sum = 410

```
#!/bin/bash
read -p "Enter number of terms and value of x : "  a  x
k=1
m=1
#sum=x
for (( i=1; i<=a; i++ ))
do
   for (( j=1; j<=k; j++ ))
   do
         m=$(( m * x ))
   done
   #echo "$m"
   if (( $(( i % 2 )) == 0 ))
   then
         m=$(( -m  ))
   fi
   echo "$m"
   sum=$(( sum + m ))
   m=1
   k=$(( k + 2 ))
done
echo " Sum : $sum "
```

Output:

```
Enter number of terms and value of x : 5 2
2
```

-8
32
-128
512
Sum : 410

---

25. Write a Shell Script that displays the n terms of square natural numbers and their sum.
1 4 9 16 ... n Terms
Test Data :
Input the number of terms : 5
Expected Output :
The square natural upto 5 terms are :1 4 9 16 25
The Sum of Square Natural Number upto 5 terms = 55

```
#!/bin/bash
read -p "Enter number of terms : " a
m=1
for (( i=1; i<=a; i++ ))
do
   for (( j=1; j<=2; j++ ))
   do
         m=$(( m * i ))
   done
   echo -n " $m "
   sum=$(( sum + m ))
   m=1
done
echo "Sum is : $sum "
```

Output:

Enter number of terms : 5
1 4 9 16 25 Sum is : 55

26. Write a Shell Script to find the sum of the series 1 +11 + 111 + 1111 + .. n terms.
Test Data :
Input the number of terms : 5
Expected Output :
1 + 11 + 111 + 1111 + 11111
The Sum is : 12345

```
#!/bin/bash
read -p "Enter a number of terms : " num
l=1
echo  "Numbers are : "
for (( i<0; i<num; i++ ))
do
        for (( j=0, k=1; j<=i; j++ ))
        do
                k=$(( 10 * k ))
        done
        echo -n "$l  "
        l=$(( k + l ))
        sum=$(( sum + l ))
#       echo -e "\n"
done

echo "Sum  : $sum"
```

Output:

```
Enter a number of terms : 5
Numbers are :
1  11  111  1111  11111  Sum  : 123455
```

---

27. Write a Shell Script to check whether a given number is a 'Perfect' number or not.
Test Data :
Input the number : 56
Expected Output :
The positive divisor : 1 2 4 7 8 14 28
The sum of the divisor is : 64
So, the number is not perfect.

```
@#! /bin/bash
```

```
read -p " enter the no " n
sum=0
for (( i=1;i<n;i++))
do
if (( n%i == 0 ))
then
echo -n " $i "
sum=`expr $sum + $i `
fi
Done
echo
echo " sum = $sum "
```

Output:

```
enter the no 56
1 2 4 7 8 14 28
sum = 64
```

---

28. Write a Shell Script to find the 'Perfect' numbers within a given number of ranges.
Test Data :
Input the starting range or number : 1
Input the ending range of number : 50
Expected Output :
The Perfect numbers within the given range : 6 28

```
#! /bin/bash
read -p " enter the starting " n
read -p " enter the no ending " m
for (( i=n ; i<=m;i++))
do
sum=0
for (( j=1 ;j<i;j++ ))
do
if (( i%j == 0))
then
sum=`expr $sum + $j`
fi
done
if (( sum == i ))
then
```

```
echo -n " $i "
fi
done
Echo
```

Output:

```
enter the starting 1
enter the no ending 56
6 28
```

---

29. Write a Shell Script to check whether a given number is an Armstrong number or not.
Test Data :
Input a number: 153
Expected Output :
153 is an Armstrong number.

```
#!/bin/bash
read -p "Enter a number : " n
num=$n
temp=$num
count=0
sum=0
temprem=1


while (( $temp > 0 ))
do
   temp=$(( temp / 10 ))
   count=$((count +1))
done


while (( $num > 0 ))
do
   rem=$(( num % 10 ))
   for (( j=1; j<=count; j++ ))
   do
         temprem=$(( temprem * rem ))
   done
   sum=$(( sum + temprem ))
   temprem=1
```

```
    num=$(( num / 10 ))
done


if [ $n -eq $sum ]
then
    echo " $n is armstrong number "
else
    echo " $n is not armstrong number "
fi
```

Output:

```
Enter a number : 153
 153 is armstrong number
```

---

30. Write a Shell Script to find the Armstrong number for a given range of number.
Test Data :
Input starting number of range: 1
Input ending number of range : 1000
Expected Output :
Armstrong numbers in given range are: 1 153 370 371 407

```
#!/bin/bash
read -p "Enter a number range  " n o
while (( $n <= $o ))
do
num=$n
temp=$num
temprem=1


while (( $temp > 0 ))
do
        temp=$(( temp / 10 ))
        count=$((count +1))
done


while (( $num > 0 ))
do
        rem=$(( num % 10 ))
```

```
        for (( j=1; j<=count; j++ ))
        do
                temprem=$(( temprem * rem ))
        done
        sum=$(( sum + temprem ))
        temprem=1
        num=$(( num / 10 ))
done


if [ $n -eq $sum ]
then
        echo " $n is armstrong number "
fi
n=$(( n + 1 ))
sum=0
count=0
Done
```

Output:

```
Enter a number range  1 160
 1 is armstrong number
 2 is armstrong number
 3 is armstrong number
 4 is armstrong number
 5 is armstrong number
 6 is armstrong number
 7 is armstrong number
 8 is armstrong number
 9 is armstrong number
 153 is armstrong number
```

---

31. Write a Shell Script to display a pattern like a diamond.

```
    *
   ***
  *****
 *******
*********
 *******
  *****
```

***
*

```
#! /bin/bash
read -p " read the no " n
m=$n
t=1
for (( i=1;i<=n;i++))
do
for (( j=1;j<=n-i;j++))
do
echo -n " "
done
for ((j=1;j<=t;j++))
do
echo -n "*"
done
echo
t=$((t+2))
done
t=$((t-4))
for ((k=1;k<m;k++))
do
for ((p=1; p<=k;p++))
do
echo -n " "
done
for ((j=1;j<=t;j++))
do
echo -n "*"
done
echo
t=$((t-2))
Done
```

---

32. Write a Shell Script to determine whether a given number is prime or not.
Test Data :
Input a number: 13
Expected Output :
13 is a prime number.

```
#!/bin/bash
```

```
read -p "Enter number of terms : " a
for (( i=1; i<=a; i++ ))
do
   if [ $(( a % i )) -eq 0 ]
   then
          echo -n "$i "
          sum=$(( sum + i ))
   fi
done

if [ $sum -gt $(( a +1 )) ]
then
   echo " Number is not Prime "
else
   echo "Number is Prime "
Fi
```

Output:

```
Enter number of terms : 43
 Number is Prime
```

---

33. Write a Shell Script to display Pascal's triangle.
Test Data :
Input number of rows: 5
Expected Output :

```
      1
    1   1
  1   2   1
 1   3   3   1
1   4   6   4   1
```

```
#! /bin/bash
read -p " enter the no " n
n=$((n+1))
for (( i=0;i<=n;i++))
do
          c=0
          for ((j=1;j<=n-i;j++))
          do
```

```
        echo -n " "
        done
        for ((k=0;k<=i;k++))
        do
        if (( k==0 || i==0 ))
        then
                c=1
        else
        c=$(( ( c * ( i-k+1 ) ) / k ))
        fi
        echo -n "$c "
done
Echo “ “
done
```

---

34. Write a Shell Script to find the prime numbers within a range of numbers.
Test Data :
Input starting number of range: 1
Input ending number of range : 50
Expected Output :
The prime number between 1 and 50 are :
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

```
#! /bin/bash
read -p " enter the starting no " m
read -p " enter the starting no " n
for ((i=m;i<=n;i++))
do
f=0
for (( j=2 ; j<i;j++))
do
if (( i%j==0))
then
f=1
fi
done
if (( f==0))
then
echo -n " $i"
fi
done
```

Output:

enter the starting no 1
enter the starting no 50
1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

---

35. Write a Shell Script to display the first n terms of the Fibonacci series.
Fibonacci series 0 1 2 3 5 8 13 .....
Test Data :
Input number of terms to display : 10
Expected Output :
Here is the Fibonacci series upto to 10 terms :
0 1 1 2 3 5 8 13 21 34

```
#! /bin/bash
read -p " enter the no " n
i=0
j=1
m=1
echo -n "$i "
echo -n $j
for (( k=2 ;k<n;k++))
do
m=$((i+j))
i=$((j))
j=$((m))
echo -n " $m"
done
echo
```

Output:

enter the no 10
0 1 1 2 3 5 8 13 21 34

---

36. Write a Shell Script to display a such a pattern for n rows using a number that starts with 1 and each row will have a 1 as the first and last number.

1
121

12321

```
#!/bin/bash
l=1
sum=0
for (( i=1; i<5; i++ ))
do
   for (( j=1; j<5-i; j++ ))
   do
         echo -n "   "
   done
   sum=1
   num=$(( $(( i / 2)) +1 ))
   for (( k=1; k<=l; k++ ))
   do
         if (( k <= num  ))
         then
               echo -n " $sum "
               sum=$(( sum + 1 ))
         else
               echo -n " $sum "
               sum=$(( sum -1 ))
         fi
   done
   l=$(( l + 2 ))
   echo -e "\n"
done
```

---

37. Write a Shell Script to display the number in reverse order.
Test Data :
Input a number: 12345
Expected Output :
The number in reverse order is : 54321

```
#!/bin/bash
read -p "Enter a number : " num
temp=$num
rev=0
while (( temp > 0 ))
do
   rem=$(( temp % 10 ))
   rev=$(( rev * 10 + rem ))
```

```
    temp=$(( temp / 10 ))
done
echo " $rev "
```

Output:

```
Enter a number : 14523
32541
```

---

38. Write a Shell Script to check whether a number is a palindrome or not.
Test Data :
Input a number: 121
Expected Output :
121 is a palindrome number.

```
#!/bin/bash
read -p "Enter a number : " num
n=$num
temp=$num
rev=0
while (( temp > 0 ))
do
    rem=$(( temp % 10 ))
    rev=$(( rev * 10 + rem ))
    temp=$(( temp / 10 ))
done
if (( n == rev ))
then

    echo "NUmber is Palindrom : $rev "
else
    echo "Number is not pallindrom : $n"
Fi
```

Output :

```
Enter a number : 145236
Number is not pallindrom : 145236
```

39. Write a Shell Script to find the number and sum of all integers between 100 and 200 which are divisible by 9.
Expected Output :
Numbers between 100 and 200, divisible by 9 :
108 117 126 135 144 153 162 171 180 189 198
The sum : 1683

```
#!/bin/bash
read -p "Enter a range : " a b
echo "NUmber between $a and $b divisible by 9 are : "
while (( a < b ))
do
   if (( $((a % 9 )) == 0 ))
   then
         echo -n " $a"
   fi
   a=$(( a + 1))
Done
```

Output:

```
Enter a range : 9 200
NUmber between 9 and 200 divisible by 9 are :
 9 18 27 36 45 54 63 72 81 90 99 108 117 126 135 144 153 162 171 180 189 198
```

---

40. Write a Shell Script to display the pyramid pattern using the alphabet.

```
      A
    A B A
  A B C B A
A B C D C B A
```

```
#! /bin/bash
read -p " enter the row " n
c=1
a=65
for ((i=1;i<=n;i++))
do
for ((j=1;j<=n-i;j++))
do
echo -n " "
```

```
done
for ((k=0;k<=c/2;k++))
do
echo $a | awk '{printf("%c",$1)}'
a=$((a+1))
done
a=$((a-2))
for ((k=0;k<c/2;k++))
do
echo $a | awk '{printf("%c",$1)}'
a=$((a-1))
done
c=$((c+2))
a=65
echo
done
```