QA. Create a basic calculator with using case.

#!/bin/bash

a=0

while (( $a!=6))

do

read -p "Enter the operation to be performed:- 1.Add 2.Sub 3.Mul 4.Div 5.Mod 6.To Exit" input

case $input in

"1")

echo "Enter A:- "

read a

echo "Enter B:- "

read b

echo "Addition:- "`expr $a + $b`

;;

"2")

echo "Enter A:- "

read a

echo "Enter B:- "

read b

echo "Substraction:- "`expr $a - $b`

;;

"3")

echo "Enter A:- "

read a

echo "Enter B:- "

read b

echo "Multiplication:- "`expr $a \\* $b`

;;

"4")

echo "Enter A:- "

read a

echo "Enter B:- "

read b

echo "Divison:- "`expr $a / $b`

;;

"5")

echo "Enter A:- "

read a

echo "Enter B:- "

read b

echo "Modulus:- "`expr $a % $b`

;;

"6")

exit

break;

;;

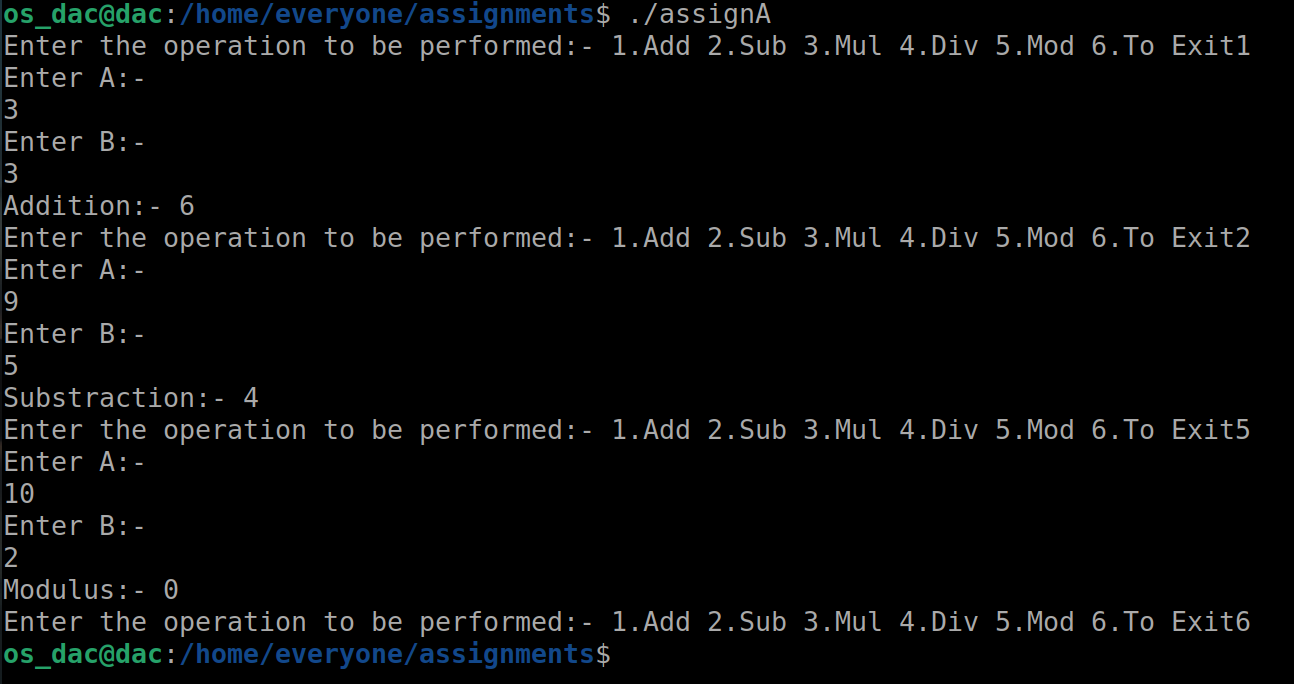
"\*")

echo "Invalid Operation"

;;

esac

done



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

#!/bin/bash

read -p "Enter A " a

echo $a

read -p "Enter B " b

echo $b

read -p "Enter C " c

echo $c

if (( $a > $b && $a > $c ))

then

echo $a "is greatest."

elif (( $b > $c ))

then

echo $b "is greatest."

else

echo $c "is greatest."

fi



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

#!/bin/bash

read -p "Enter the Number of files to be created " a

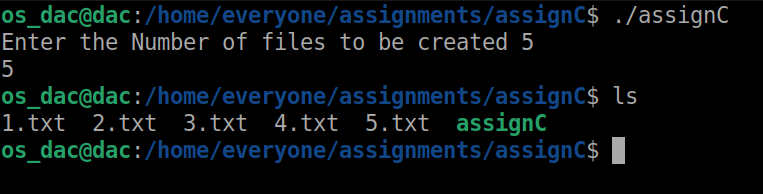
echo $a

for (( i=1; i<=$a; i++ ))

do

touch $i.txt

done



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

#!/bin/bash

read -p "Enter Number to check it is even or odd " num

echo $num

if (( `expr $num%2==0` ))

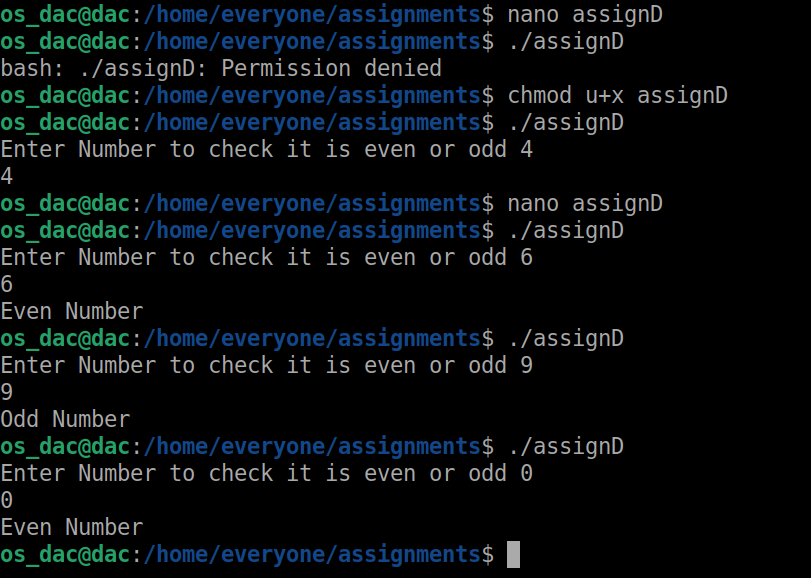
then

echo "Even Number"

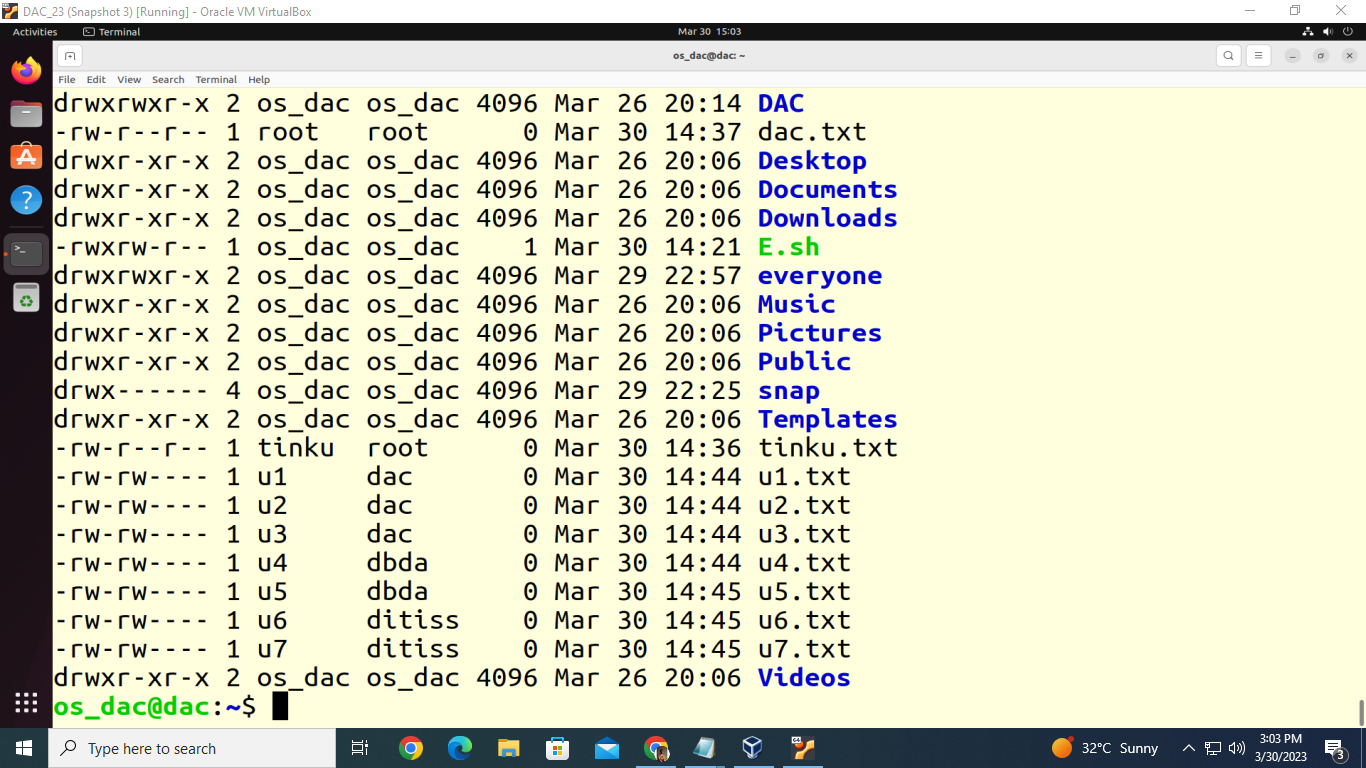
else

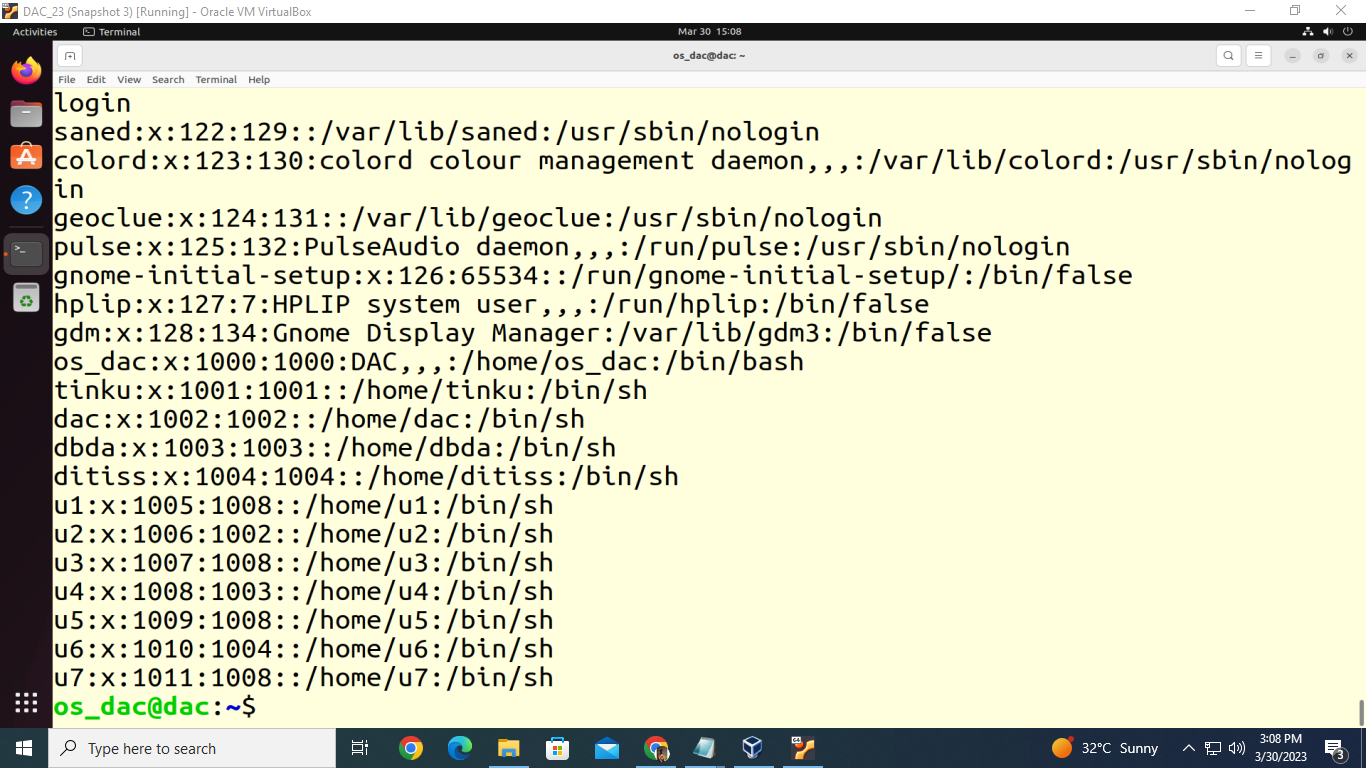
echo "Odd Number"

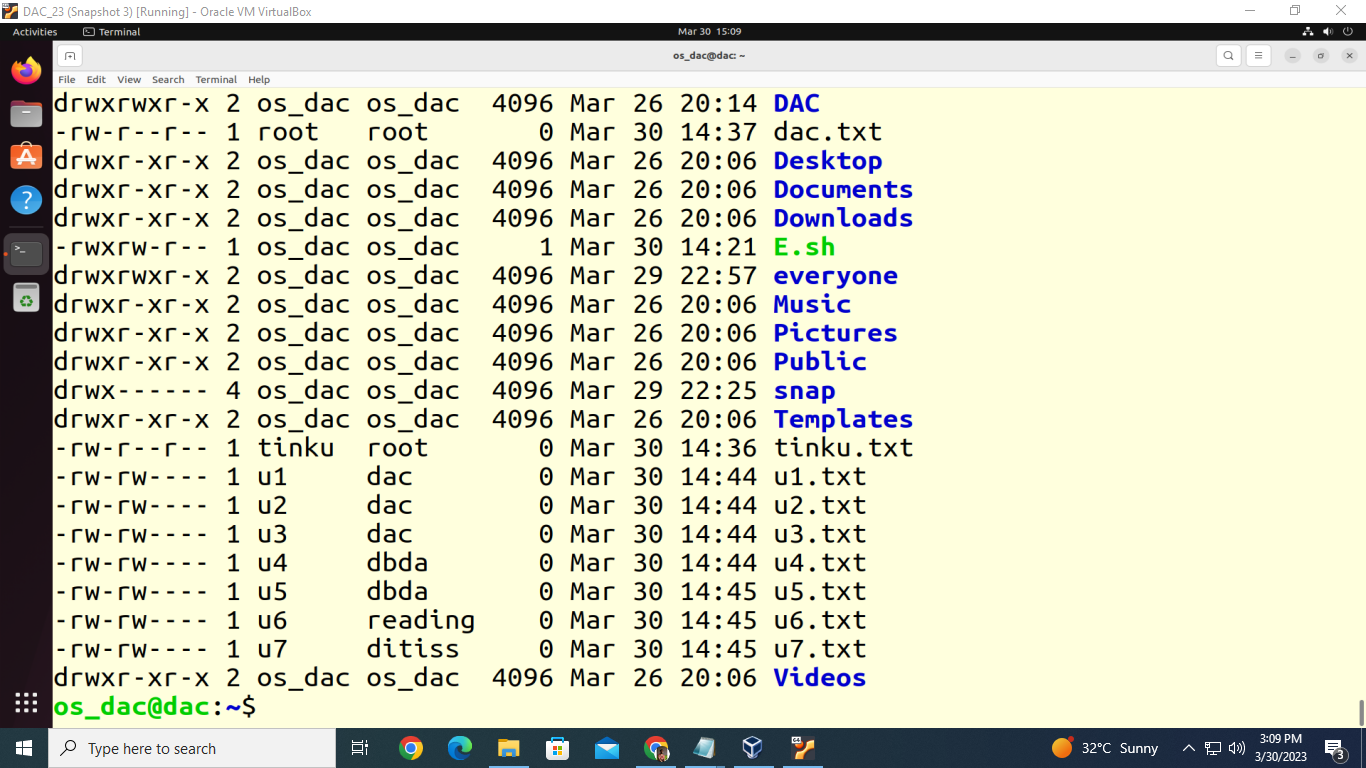
fi



QE. Follow the group file and complete the tasks mentioned inside it.







Q1. Write a Shell Script to display the first 10 natural numbers.

#!/bin/bash

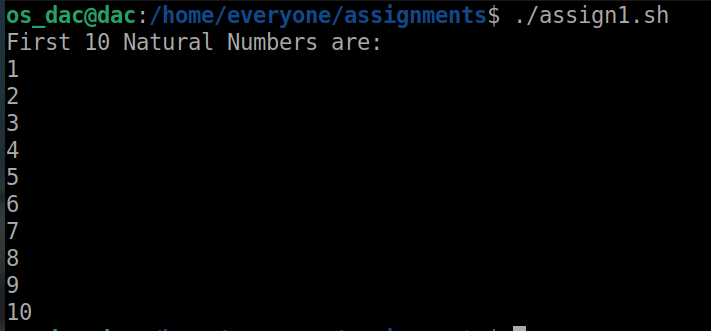
echo "First 10 Natural Numbers are: "

for i in {1..10}

do

echo $i

done



Q2. Write a Shell Script to compute the sum of the first 10 natural numbers.

#!/bin/bash

echo "Sum of first 10 natural numbers is"

sum=0

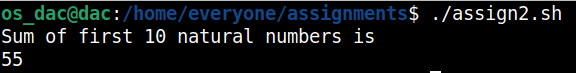
for ((i=1; i<=10; i++))

do

sum=`expr $sum + $i`

done

echo $sum



Q3. 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

read -p "Enter " num

echo "The first "$num" natural numbers are "

for ((i=1; i<=$num; i++))

do

echo $i

done

echo "Sum of "$num" natural numbers is "

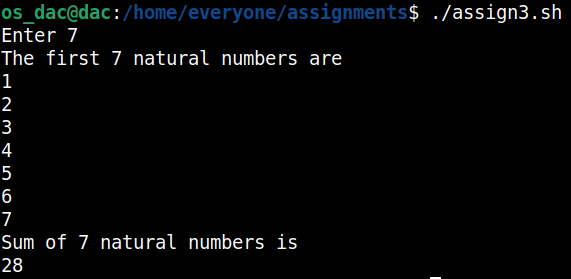
sum=0

for ((i=1; i<=$num; i++))

do

sum=`expr $sum + $i`

done



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

#!/bin/bash

echo "Enter 10 Numbers"

sum=0

for ((i=0; i<10; i++))

do

read -p "Enter Number " num

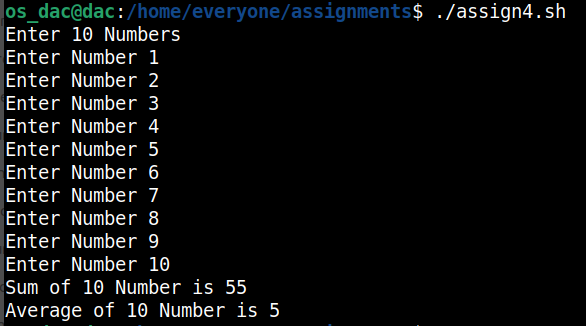
sum=`expr $sum + $num`

avg=`expr $sum / 10`

done

echo "Sum of 10 Number is "$sum

echo "Average of 10 Number is "$avg



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

read -p "Enter the Number to find Cube upto " num

cube=1

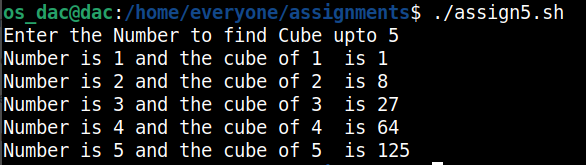
for((i=1; i<=$num; i++))

do

cube=`expr $i \\* $i \\* $i`

echo "Number is "$i "and the cube of "$i " is "$cube

done



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

#!/bin/bash

read -p "Input the Number (Table to be calculated) " num

mul=1

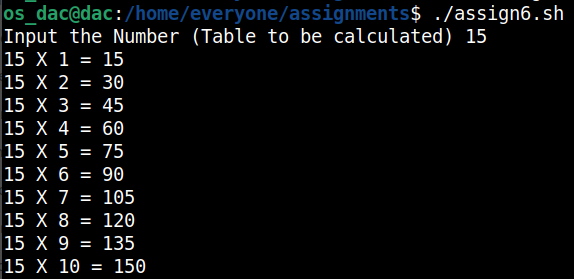
for((i=1; i<=10; i++))

do

mul=`expr $num \\* $i`

echo $num "X" $i "=" $mul

done



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 Input number starting from 1 to n" num

mult=1

for((i=1; i<=$num; i++))

do

for((j=1; j<=10; j++))

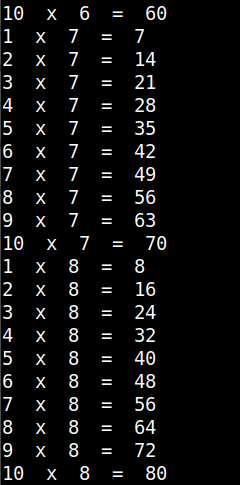
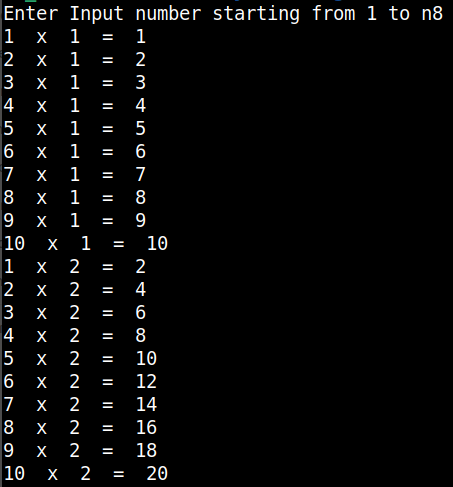
do

mult=`expr $i \\* $j`

echo $j " x " $i " = " $mult

done

done



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

#!/bin/bash

read -p "Enter Input " num

sum=0

for ((i=1; i<=$num \\* 2; i++))

do

if(( $i % 2!=0 ))

then

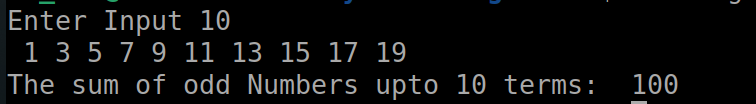
echo -n " "$i

sum=`expr $sum + $i`

fi

done

echo -e "\nThe sum of odd Numbers upto "$num" terms: " $sum



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

The pattern like :

\*

\*\*

\*\*\*

\*\*\*\*

#!/bin/bash

read -p "Enter Number to print star pattern " num

for ((i=0; i<=$num; i++))

do

for((j=0; j<=$i; j++))

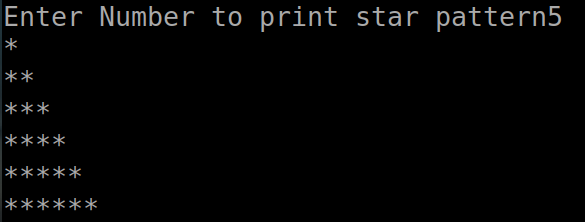
do

echo -n "\*"

done

echo

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

read -p "Enter Number to print star pattern " num

for ((i=1; i<=$num; i++))

do

for((j=1; j<=$i; j++))

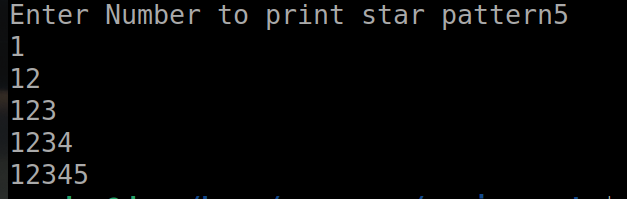
do

echo -n "$j"

done

echo

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

read -p "Enter Number to print star pattern " num

for ((i=1; i<=$num; i++))

do

for((j=1; j<=$i; j++))

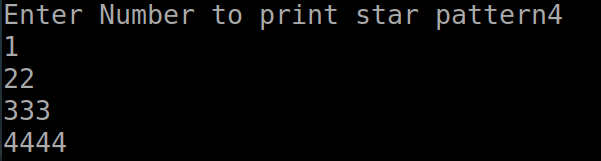
do

echo -n "$i"

done

echo

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=0

for ((1=1;i<=5;i++))

do

for ((j=1;j<$i;j++))

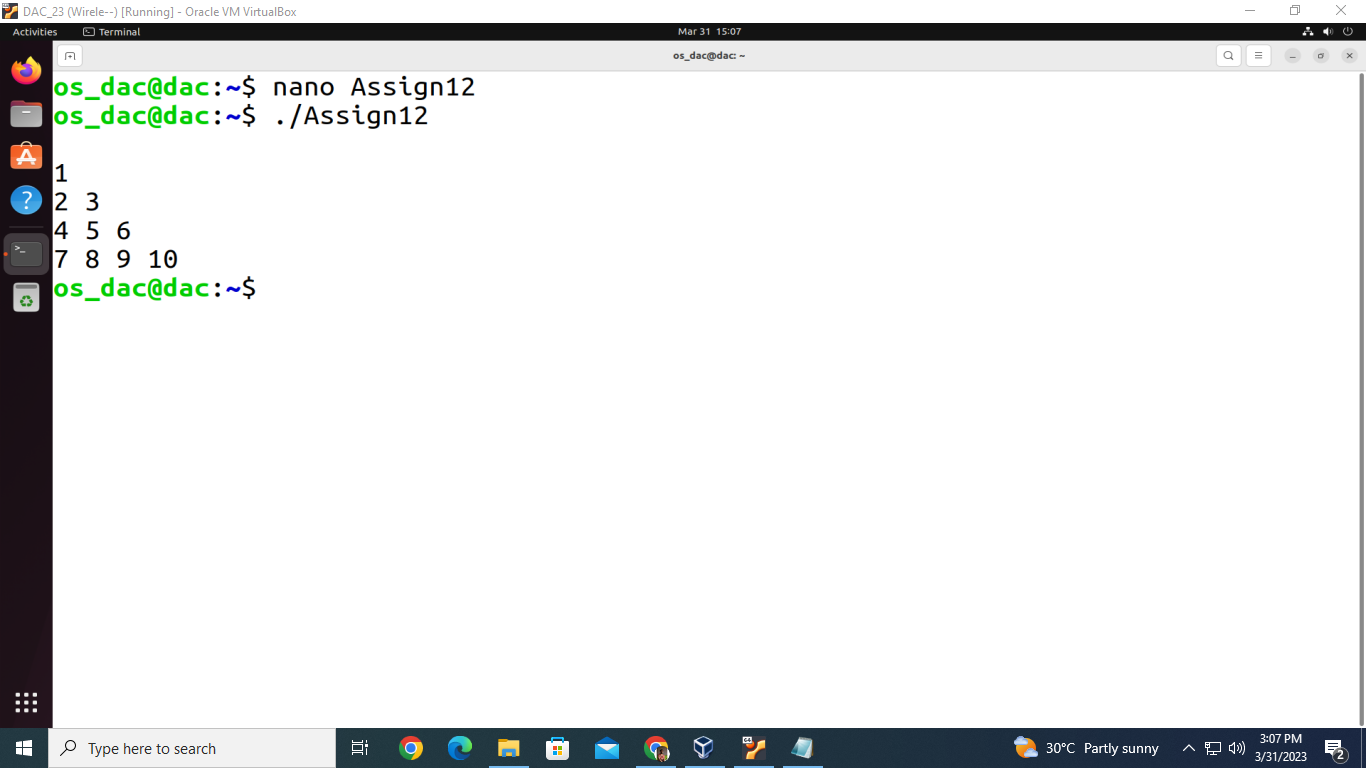
do

k=$((k + 1)) echo -n "$k"

done

echo " "

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

read -p "Input number of rows : " rows

spc=$(( rows + 4 - 1 ))

t=1

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

do

for (( k=spc; k>=1; k-- ))

do

echo -n " "

done

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

do

echo -n "$t "

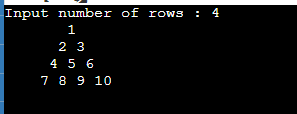
(( t++ ))

done

echo ""

(( spc-- ))

done



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

\*

\* \*

\* \* \*

\* \* \* \*

#!/bin/bash

k=0

s=4

for((i=1;i<5;i++))

do

echo -n " "

done

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

do

k=$((k+1))

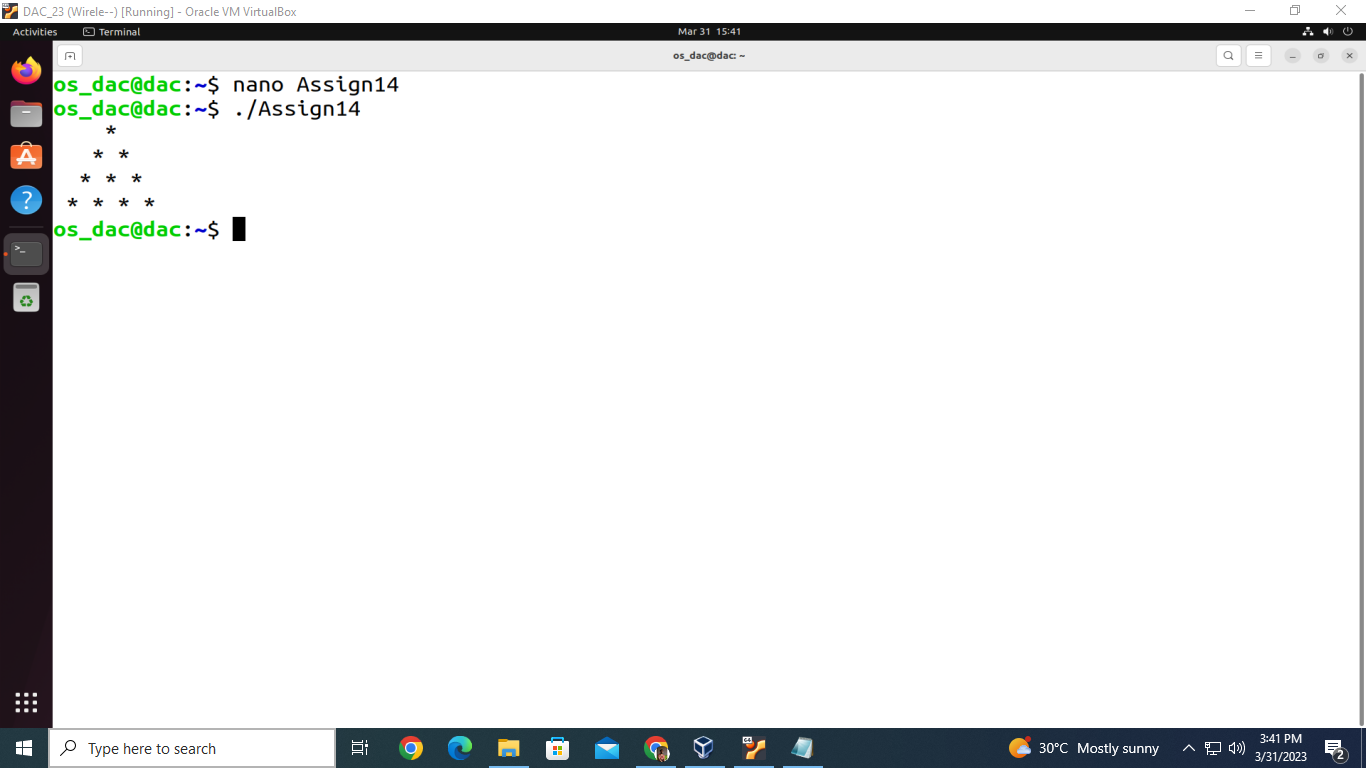
echo -n "\* "

done

echo " "

s=$((s - 1))

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 the Number for which you want factorial " num

mult=1

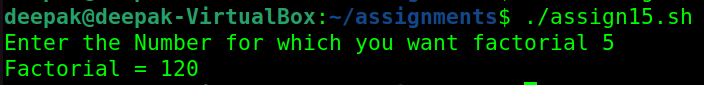
for((i=1;i<=num;i++))

do

mult=$((mult \* i))

done

echo "Factorial = "$mult



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 Number upto which you want even numbers " num

sum=0

for((i=2;i<=num \* 2;i++))

do

if((i % 2 == 0))

then

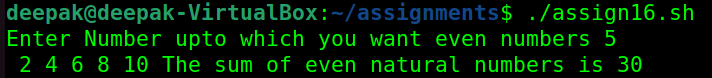
echo -n " "$i

sum=$((sum + i))

fi

done

echo " The sum of even natural numbers is "$sum



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

k=0

s=4

for((i=1;i<5;i++))

do

echo -n " "

done

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

do

k=$((k+1))

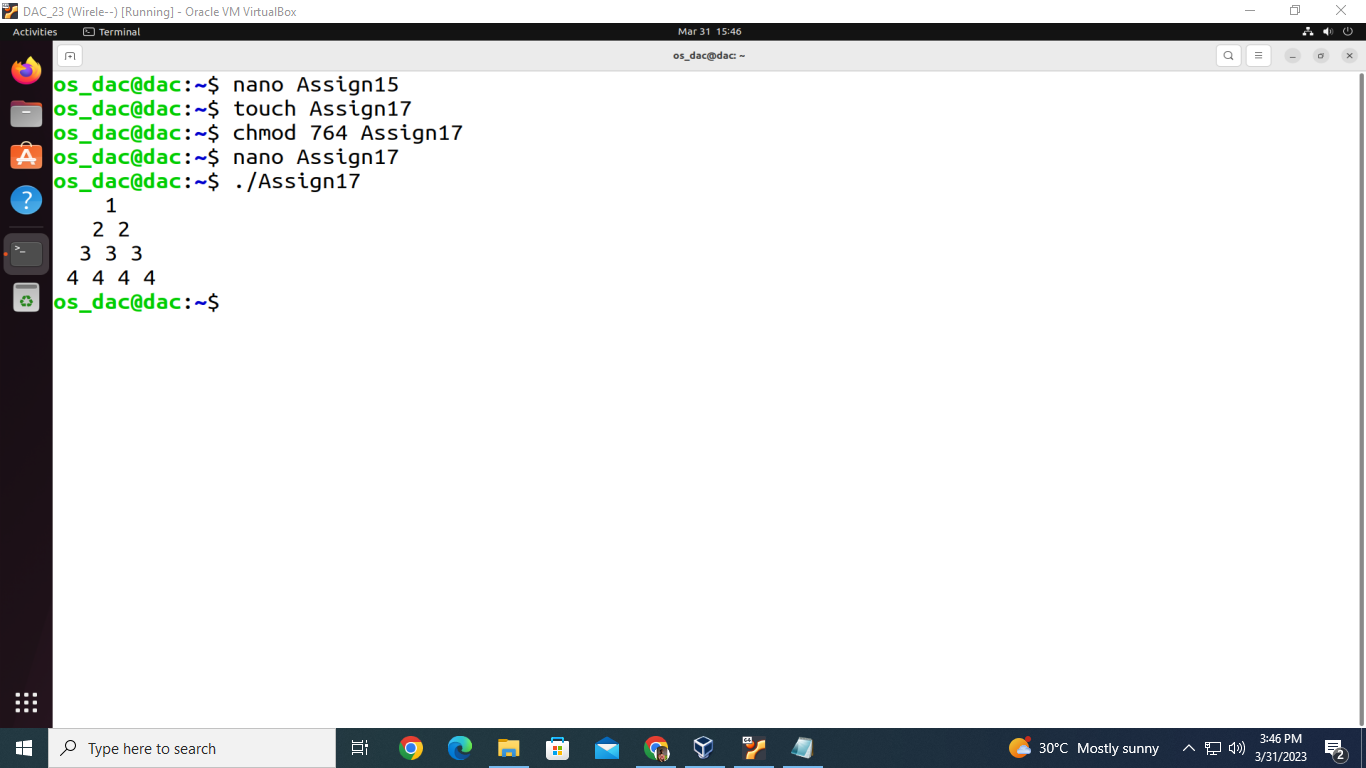
echo -n "$i "

done

echo " "

s=$((s - 1))

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 "Input the Value of x :" x

read -p "Input the number of terms : " n

sum=1

t=1

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

do

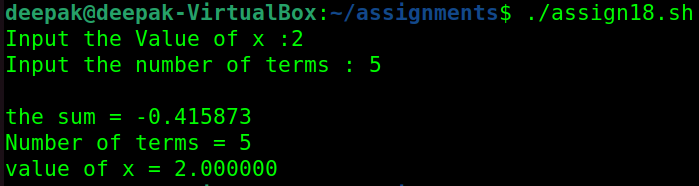
d=$(( 2\*i\*(2\*i-1) ))

t=$(bc -l <<< "-($t\*$x\*$x)/$d")

sum=$(bc -l <<< "$sum+$t")

done

printf "\nthe sum = %f\nNumber of terms = %d\nvalue of x = %f\n" "$sum" "$n" "$x"



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

i=1

s=0.0

echo -n "Input the number of terms : "

read n

echo ""

while [ $i -le $n ]

do

if [ $i -lt $n ]

then

echo -n "1/$i + "

s=$(echo "scale=5;$s + 1/$i" | bc)

elif [ $i -eq $n ]

then

echo -n "1/$i"

s=$(echo "scale=5;$s + 1/$i" | bc)

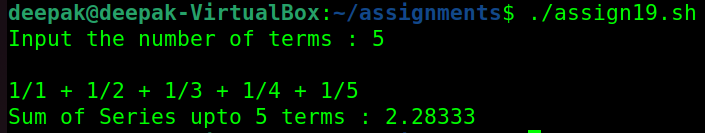
fi

i=$((i+1))

done

echo ""

echo "Sum of Series upto $n terms : $s"



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

echo "Enter the number of rows: "

read n

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

do

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

do

echo -n " "

done

for((j=1; j<=2\*i-1; j++))

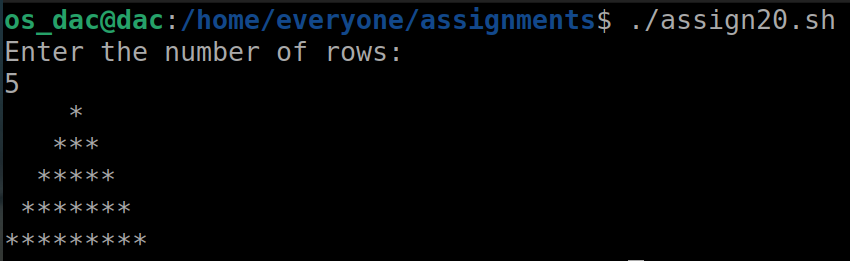
do

echo -n "\*"

done

echo ""

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 series = 111105

#!/bin/bash

t=9

sum=0

read -p "Input the number " num

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

do

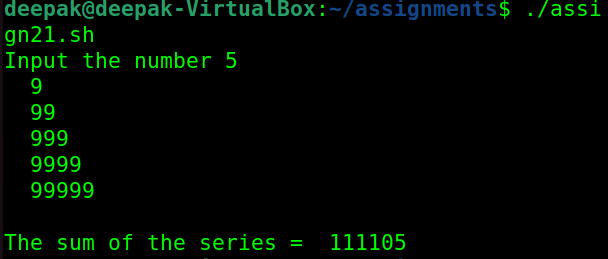
sum=$((sum+t))

echo " " $t

t=$((t\*10+9))

done

echo -e "\nThe sum of the series = " $sum



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

1

01

101

0101

10101

#!/bin/bash

read -p "Input number of rows : " n

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

do

if (( i%2 == 0 ))

then

p=1

q=0

else

p=0

q=1

fi

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

do

if (( j%2 == 0 ))

then

echo -n "$p"

else

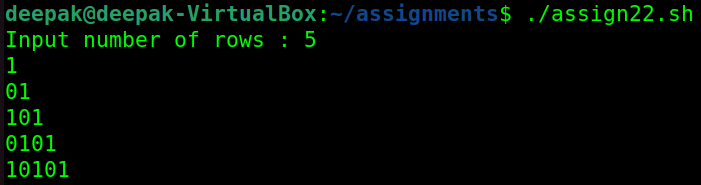
echo -n "$q"

fi

done

echo ""

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 "Input the value of x :" x

read -p "Input number of terms : " n

sum=1

no\_row=1

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

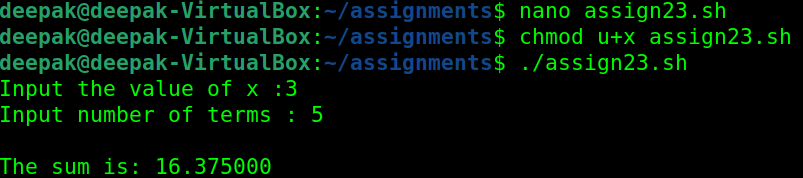
do

no\_row=$(bc -l <<< "$no\_row\*$x/$i")

sum=$(bc -l <<< "$sum+$no\_row")

done

printf "\nThe sum is: %f\n" "$sum"



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 "Input the value of x :" x

read -p "Input number of terms : " n

sum=$x

m=-1

printf "The values of the series: \n"

printf "%d\n" "$x"

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

do

ctr=$(( 2\*i+1 ))

mm=$(( x\*\*ctr ))

nn=$(( mm\*m ))

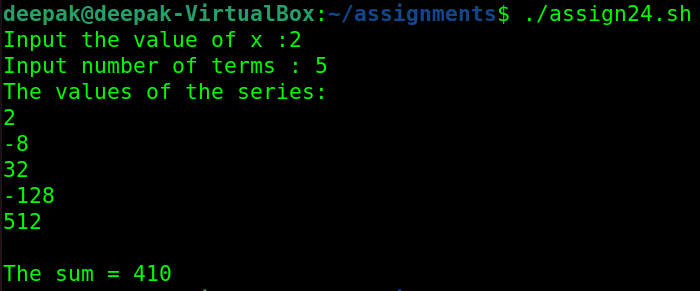
printf "%d\n" "$nn"

sum=$(( sum+nn ))

m=$(( m\*(-1) ))

done

printf "\nThe sum = %d\n" "$sum"



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 "Input the number of terms " num

sum=0

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

do

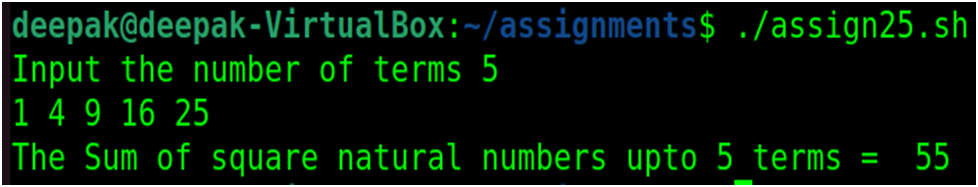
sqr=$((i\*i))

echo -n $sqr" "

sum=$((sum + sqr))

done

echo -e "\nThe Sum of square natural numbers upto "$num" terms = " $sum



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 "Input the number of terms " num

sum=0

t=1

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

do

echo -n "$t "

if((i<num))

then

echo -n "+ "

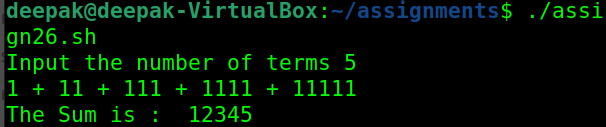
fi

sum=$((sum + t))

t=$((((t\*10))+1))

done

echo -e "\nThe Sum is : " $sum



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 "Input the number : " n

sum=0

echo -n "The positive divisor : "

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

do

if (( n%i == 0 ))

then

sum=$(( sum+i ))

echo -n "$i "

fi

done

echo

echo "The sum of the divisor is : $sum"

if (( sum == n ))

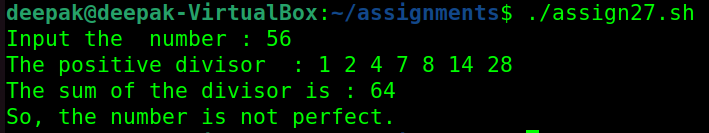
then

echo "So, the number is perfect."

else

echo "So, the number is not perfect."

fi



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

echo "Input the starting range or number : "

read mn

echo "Input the ending range of number : "

read mx

echo "The Perfect numbers within the given range : "

for ((n=$mn; n<=$mx; n++))

do

i=1

sum=0

while [ $i -lt $n ]

do

if [ $(($n%$i)) -eq 0 ]

then

sum=$(($sum+$i))

fi

i=$(($i+1))

done

if [ $sum -eq $n ]

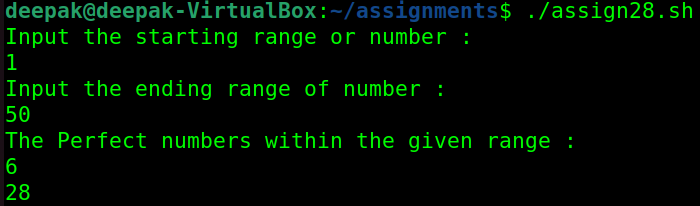
then

echo $n

fi

done

echo



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 "Input a Number " num

temp1=$num

sum=0

while((num!=0))

do

temp2=$((num % 10))

mult=$((temp2 \* temp2 \* temp2))

sum=$((mult + sum))

num=$((num / 10))

done

if (($sum == temp1 ))

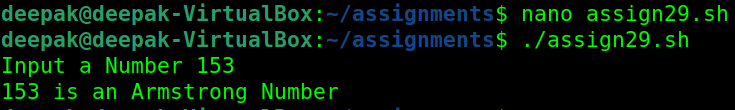
then

echo $sum "is an Armstrong Number"

else

echo $sum "is not an Armstrong Number"

fi



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 "Input starting number of range: " stno

read -p "Input ending number of range: " enno

echo -n "Armstrong numbers in given range are: "

for (( num=$stno; num<=$enno; num++ ))

do

temp=$num

sum=0

while [ $temp -ne 0 ]

do

r=$((temp % 10))

temp=$((temp / 10))

sum=$((sum + r \* r \* r))

done

if [ $sum -eq $num ]

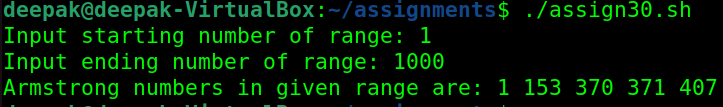
then

echo -n "$num "

fi

done

echo ""



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

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

#!/bin/bash

echo "Enter the number of rows: "

read n

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

do

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

do

echo -n " "

done

for((j=1; j<=2\*i-1; j++))

do

echo -n "\*"

done

echo ""

done

for((i=n-1; i>=1; i--))

do

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

do

echo -n " "

done

for((j=1; j<=2\*i-1; j++))

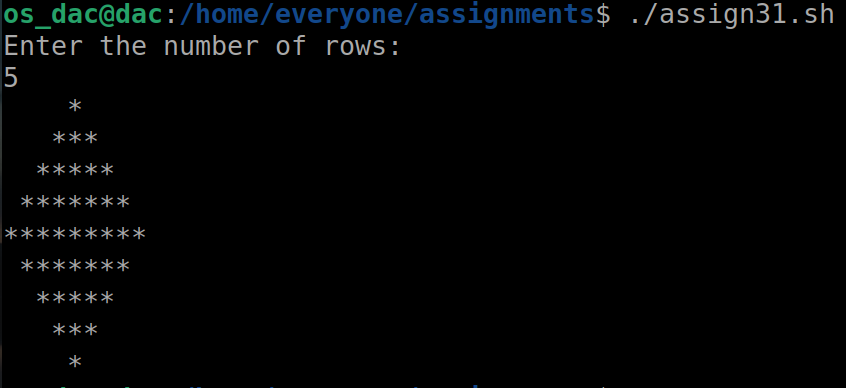
do

echo -n "\*"

done

echo ""

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 the Number you want to check is prime or not " num

for((i=2;i<num;i++))

do

if((num % i == 0))

then

flag=1

fi

done

if((flag==0))

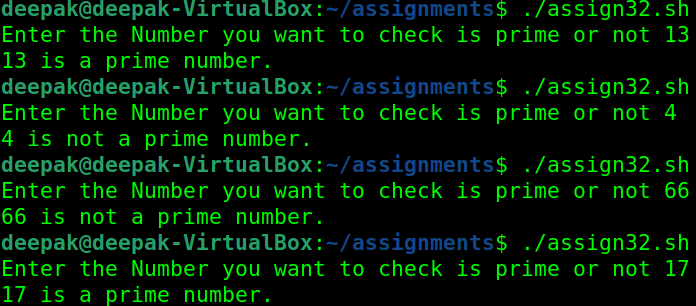
then

echo $num "is a prime number."

else

echo $num "is not a prime number."

fi



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 "Input number of rows: " no\_row

for ((i=0;i<no\_row;i++))

do

for ((blk=1;blk<=no\_row-i;blk++))

do

printf " "

done

c=1

for ((j=0;j<=i;j++))

do

if ((j==0 || i==0))

then

c=1

else

c=$((c\*(i-j+1)/j))

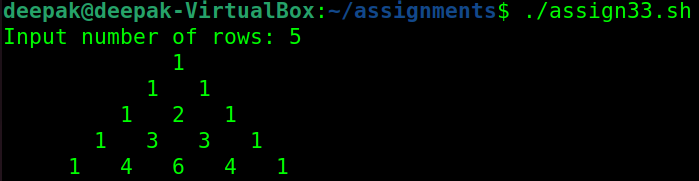
fi

printf "% 4d" $c

done

printf "\n"

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 "Input starting number of range: " stno

read -p "Input ending number of range: " enno

echo "The prime numbers between $stno and $enno are:"

for (( num=$stno; num<=$enno; num++ ))

do

ctr=0

for (( i=2; i<=$num/2; i++ ))

do

if [ $((num%i)) -eq 0 ]

then

ctr=$((ctr+1))

break

fi

done

if [ $ctr -eq 0 ] && [ $num -ne 1 ]

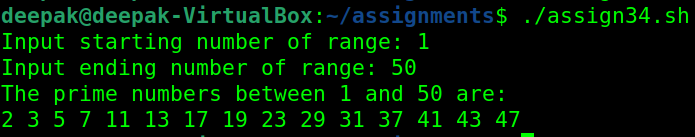
then

echo -n "$num "

fi

done

echo ""



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 "Input number of terms to display " num

prv=0

pre=1

echo -n $prv" "

echo -n $pre" "

for((i=3;i<=num;i++))

do

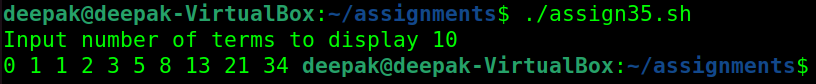
trm=$(($prv + $pre))

echo -n $trm" "

prv=$pre

pre=$trm

done



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

read -p "Input number of rows: " n

for (( i=0; i<=n; i++ ))

do

# print blank spaces

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

do

echo -n " "

done

# display number in ascending order upto middle

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

do

echo -n "$j"

done

# display number in reverse order after middle

for (( j=i-1; j>=1; j-- ))

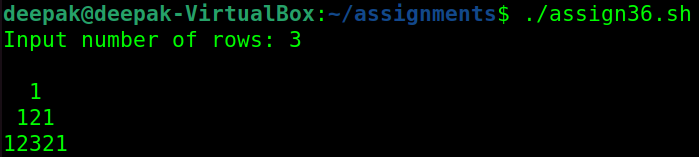
do

echo -n "$j"

done

echo ""

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 the Number you want to reverse " num

rev=0

remainder=0

while(($num>0))

do

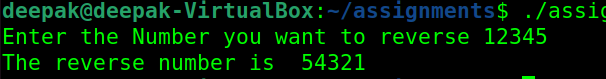
remainder=$((num % 10))

rev=$((rev \* 10 + remainder))

num=$((num / 10))

done

echo "The reverse number is " $rev



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 "Input a number: " num r t

sum=0

for((t=num;num!=0;num=num/10))

do

r=$((num%10))

sum=$((sum\*10+r))

done

if((t==sum))

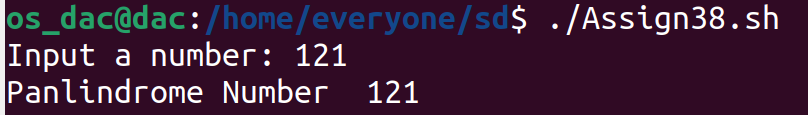
then

echo $t

else

echo "Is not Panlindrome: "$t

fi



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

echo "Enter Numbers Between 100 and 200 divisible by 9: "

sum=0

for ((i=101; i<200; i++))

do

if (($i%9==0))

then

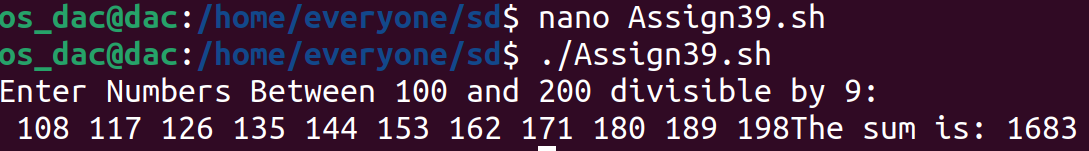
echo -n " "$i

sum=`expr $sum + $i`

fi

done

echo "The sum is: "$sum



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 "Input number of rows: " n

for (( i=0; i<=n; i++ ))

do

# print blank spaces

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

do

echo -n " "

done

# display number in ascending order upto middle

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

do

printf "\x$(printf %x $((64+j)))"

done

# display number in reverse order after middle

for (( j=i-1; j>=1; j-- ))

do

printf "\x$(printf %x $((64+j)))"

done

echo ""

done

