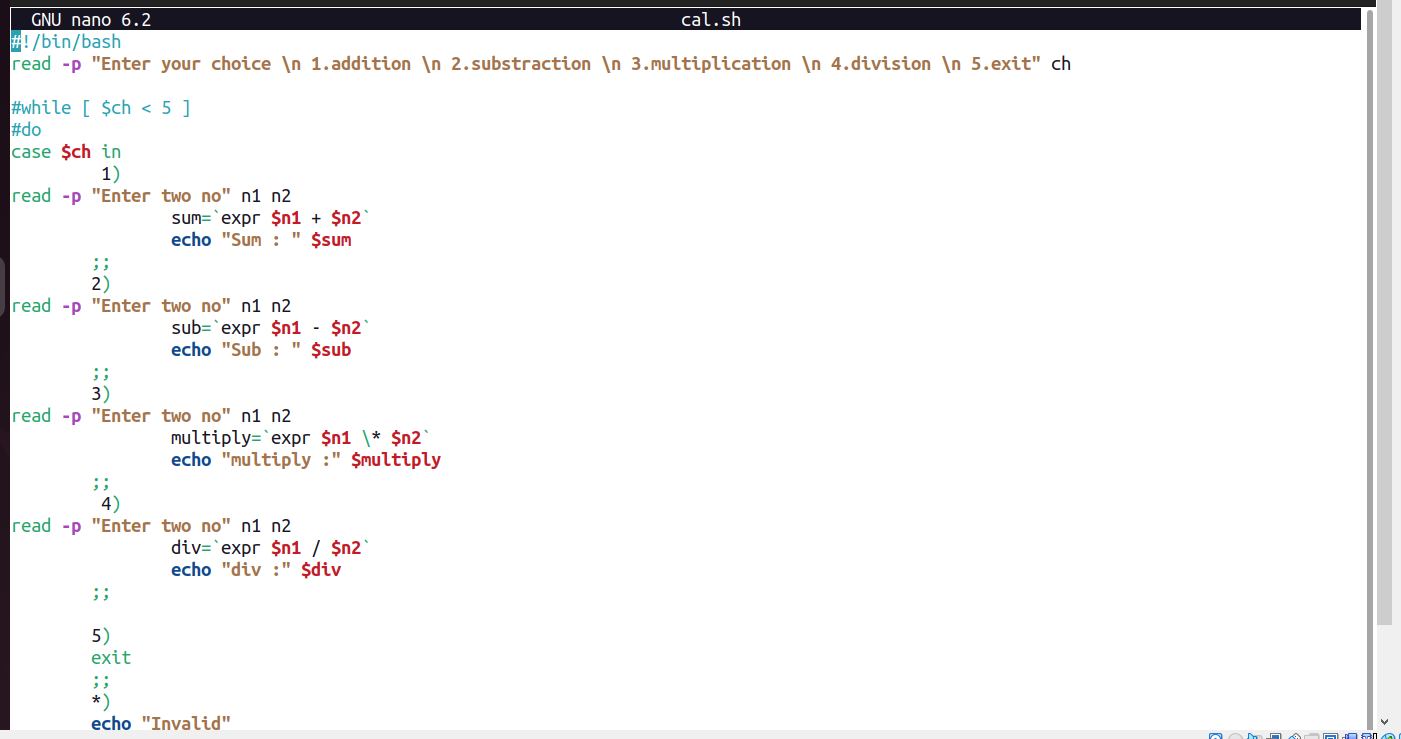
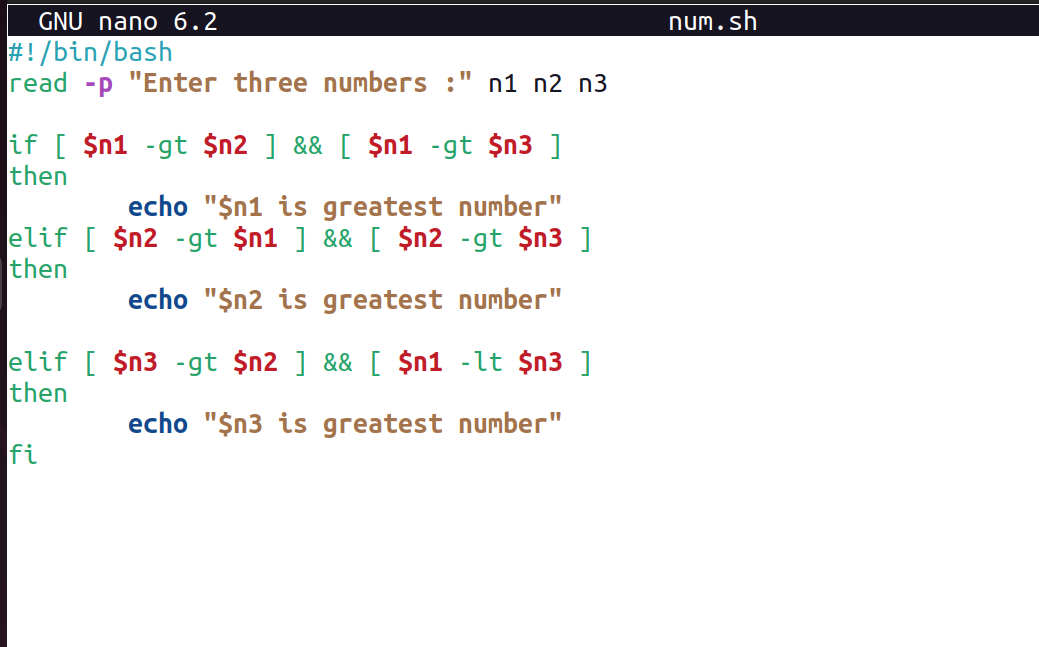
A. Create a basic calculator with using case.



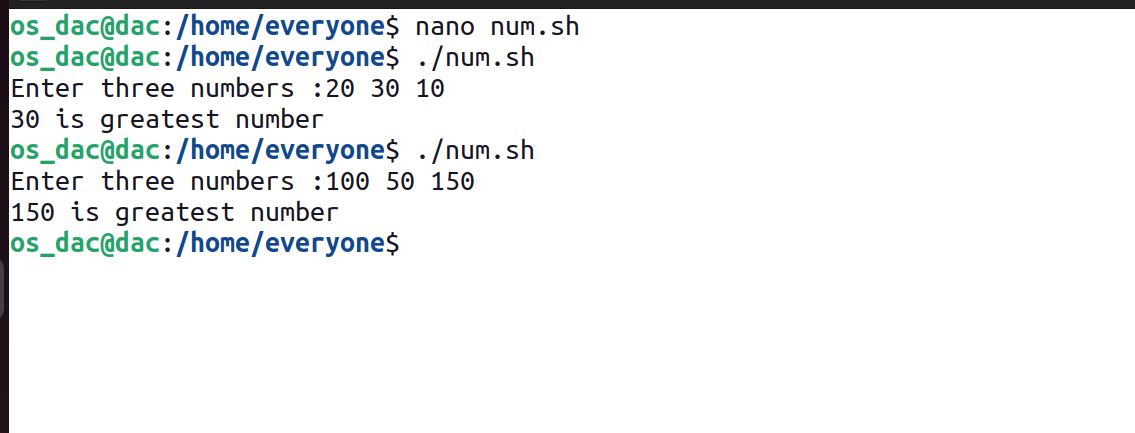
ANS:-



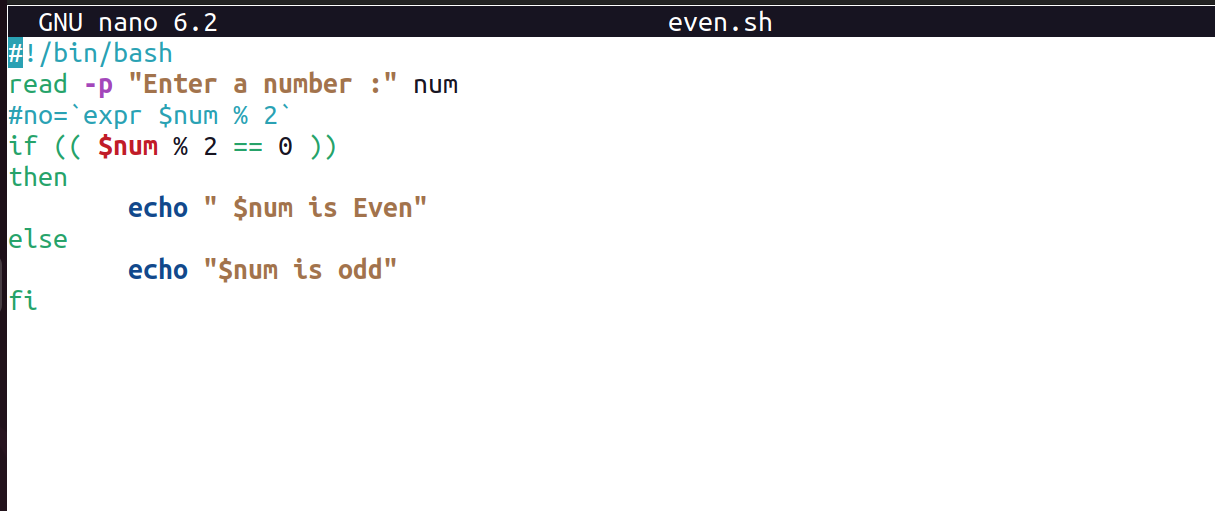
B. Find out the greatest number among three numbers entered by users using if condition



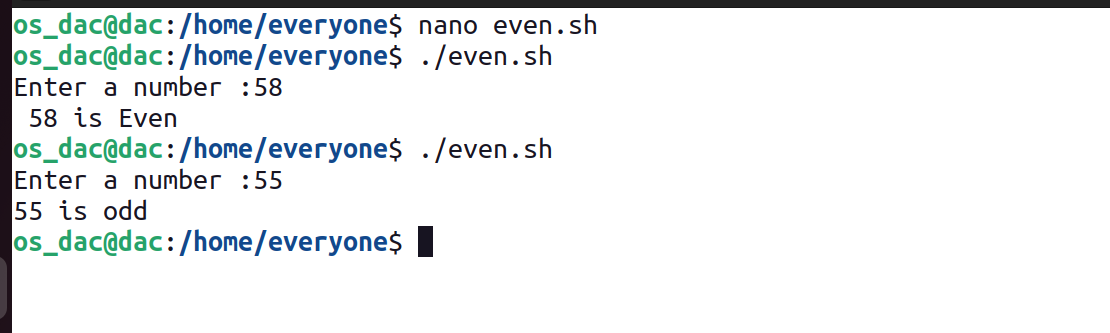
ANS:-



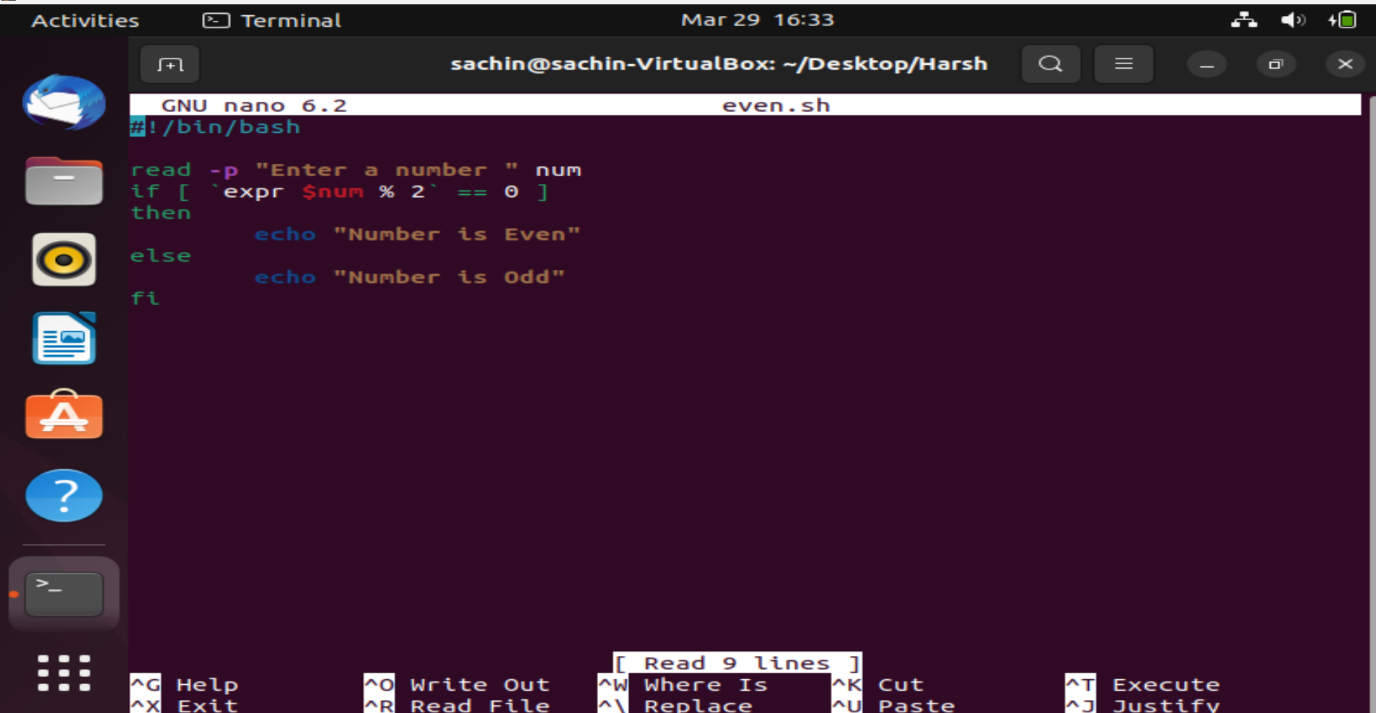
C. Write a program to take input of number from user and generate that number of .txt files.

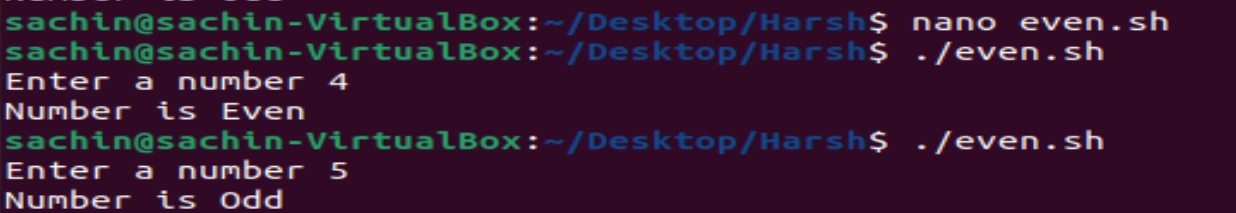


ANS:-



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





E.Follow the group file and complete the tasks inside it.

group.txt file:-

Username                      GroupName                      sudo

tinku                         tinku   y

dac                           dac       y

u1                       dac                             n

u2                            dac                             n

u3                           dac                             n

dbda                        dbda                            y

u4                          dbda                            n

U5                            dbda   n

ditiss                       ditiss   y

U6 ditiss   n

U7 ditiss   n

Create a text file with every user.

Remove the read permission for others.

Check u4.txt is readable by which users?

create a group readings and add the user u1,u3,u5 and u7 to the group.

change the group of u6.txt to reading.

asdf@asdfv:~$ cat /etc/group | grep sudo

sudo:x:27:asdf

asdf@asdfv:~$ sudo adduser tinku sudo

Adding user `tinku' to group `sudo' ...

Adding user tinku to group sudo

Done.

asdf@asdfv:~$ cat /etc/group | grep sudo

sudo:x:27:asdf,tinku

asdf@asdfv:~$ sudo useradd dac

asdf@asdfv:~$ sudo passwd dac

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ usermod -a G sudo dac

asdf@asdfv:~$ sudo usermod -aG sudo dac

asdf@asdfv:~$ sudo adduser u1 dac

Adding user `u1' to group `dac' ...

Adding user u1 to group dac

Done.

asdf@asdfv:~$ sudo useradd -G sudo -m u2

asdf@asdfv:~$ sudo passwd u2

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ cat /etc/passwd | grep u2

u2:x:1004:1004::/home/u2:/bin/sh

asdf@asdfv:~$ cat /etc/group |grep u2

sudo:x:27:asdf,tinku,dac,u2

u2:x:1004:

asdf@asdfv:~$ sudo userdel u2

asdf@asdfv:~$ sudo useradd -G dac -m u2

useradd: warning: the home directory /home/u2 already exists.

useradd: Not copying any file from skel directory into it.

asdf@asdfv:~$ cat /etc/group |grep u2

dac:x:1002:u1,u2

u2:x:1004:

asdf@asdfv:~$ #

asdf@asdfv:~$ #creating user u3 in group dac

asdf@asdfv:~$ sudo useradd -G dac -m u3

asdf@asdfv:~$ sudo passwd u1

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ sudo passwd u2

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ # creatingg user dbda and adding it in sudo group in single command

asdf@asdfv:~$ sudo useradd -G sudo -m dbda

asdf@asdfv:~$ sudo passwd dbda

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ sudo useradd -G dbda -m u4

asdf@asdfv:~$ sudo passwd u4

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ sudo useradd -G dbda -m u5

asdf@asdfv:~$ sudo passwd u5

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ sudo useradd -G sudo -m ditiss

asdf@asdfv:~$ sudo passwd ditiss

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ cat /etc/group | grep sitiss

asdf@asdfv:~$ cat /etc/group | grep ditiss

sudo:x:27:asdf,tinku,dac,dbda,ditiss

ditiss:x:1009:

asdf@asdfv:~$ sudo useradd -G ditiss -m u6

asdf@asdfv:~$ sudo useradd -G ditiss -m u7

asdf@asdfv:~$ sudo passwd u6

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$ sudo passwd u7

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: password updated successfully

asdf@asdfv:~$

dac:x:1002:u1,u2,u3,u4

dbda:x:1006:u4,u5

u4:x:1007:

$ #only u4 exits in u4 group

$ exit

asdf@asdfv:~/ass$ sudo groupadd readings

asdf@asdfv:~/ass$ sudo usermod -aG readings u1

asdf@asdfv:~/ass$ sudo usermod -aG readings u3

asdf@asdfv:~/ass$ sudo usermod -aG readings u5

asdf@asdfv:~/ass$ sudo usermod -aG readings u7

asdf@asdfv:~/ass$ cat /etc/group | grep readings

readings:x:1015:u1,u3,u5,u7

asdf@asdfv:~/ass$ su u6

Password:

$ touch u6.txt

touch: cannot touch 'u6.txt': Permission denied

$ sudo u6.txt

[sudo] password for u6:

sudo: u6.txt: command not found

$ sudo adduser u6 sudo

The user `u6' is already a member of `sudo'.

$ sudo touch u6.txt

$ ls

ass  asss  asssa  ass.sh  fin  final.txt  sad  u6.txt

$ sudo chgrp readings u6.txt

$ pwd

/home/asdf/ass

$ cd /home/u6

$ pwd

/home/u6

$ sudo touch u6.txt

$ ls -ls

total 0

0 -rw-r--r-- 1 root root 0 Apr  2 23:49 u6.txt

$ sudo chown u6 u6.txt

$ ls -ls

total 0

0 -rw-r--r-- 1 u6 root 0 Apr  2 23:49 u6.txt

$ sudo chgrp readings u6.txt

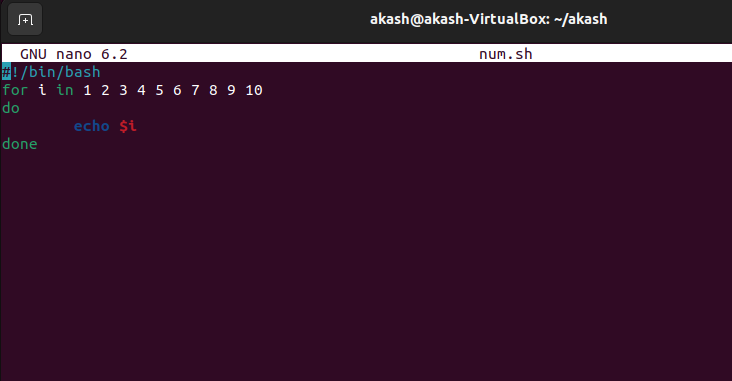
$ ls -ls

total 0

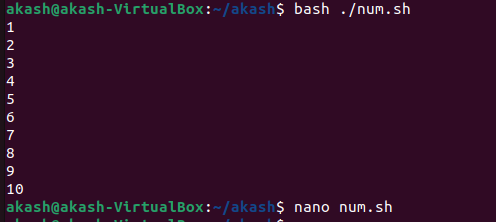
0 -rw-r--r-- 1 u6 readings 0 Apr  2 23:49 u6.txt

$

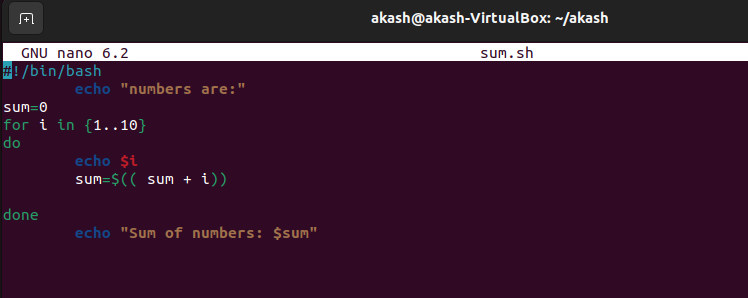
Q1. Write a Shell Script to display the first 10 natural numbers.



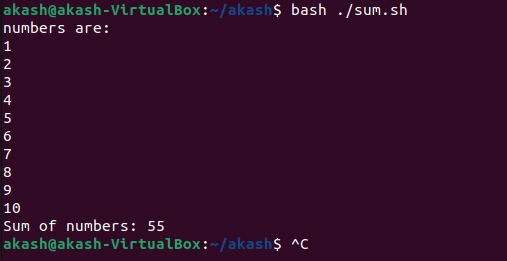
Ans-



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



Ans:-



Q3. Write a Shell Script to display n terms of natural numbers and their sum.

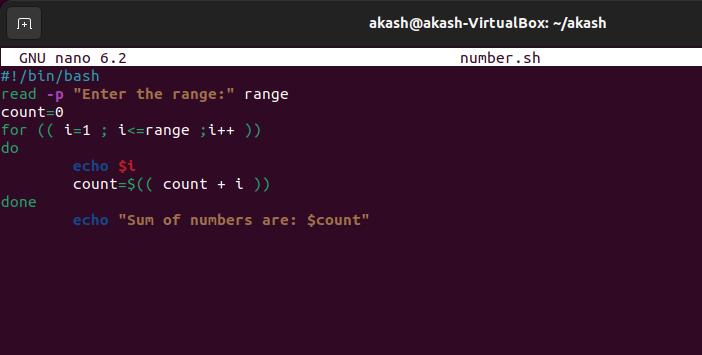
Test Data : 7

Expected Output :

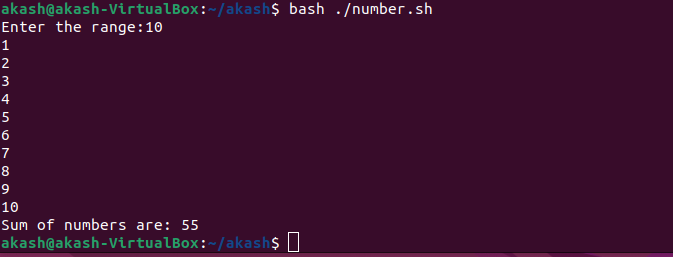
The first 7 natural number is :

1 2 3 4 5 6 7

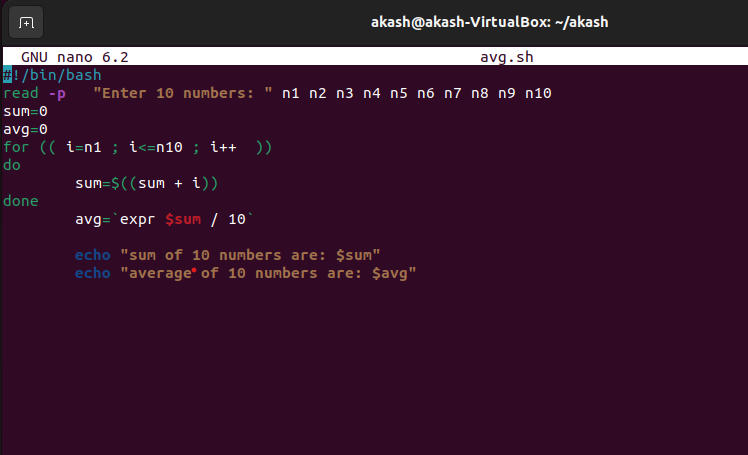
The Sum of Natural Number upto 7 terms : 28

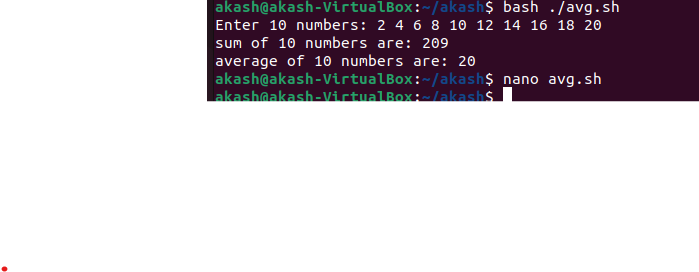


Ans:-



Q4. Write a Shell Script to read 10 numbers from the keyboard and find their sum and average.





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

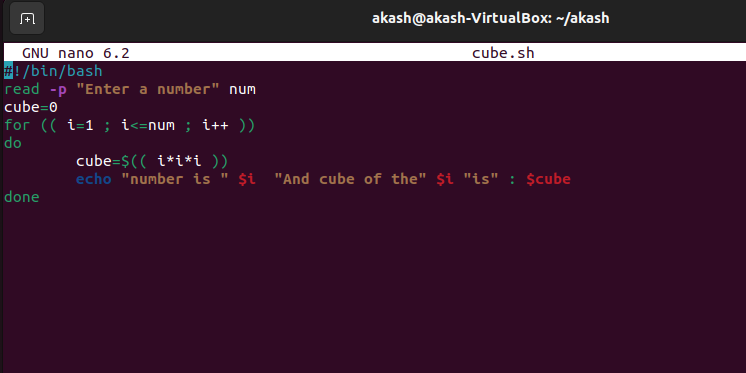
Number is : 2 and cube of the 2 is :8

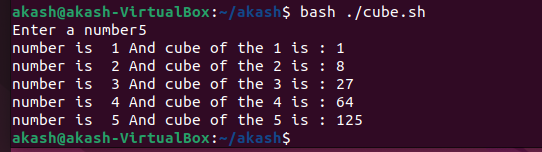
Number is : 3 and cube of the 3 is :27

Number is : 4 and cube of the 4 is :64

Number is : 5 and cube of the 5 is :125

Ans:-





Q6. 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

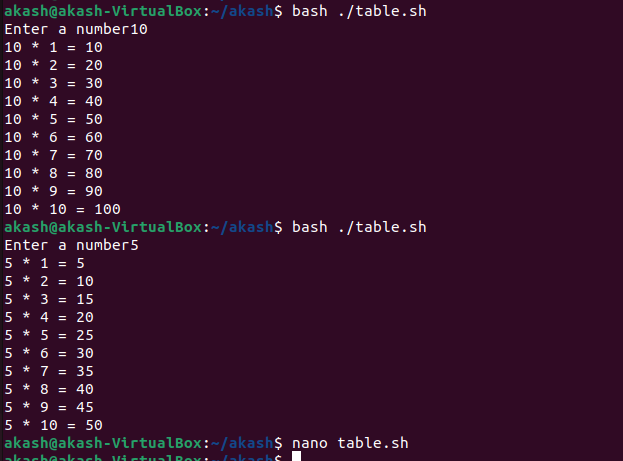
...

...

15 X 10 = 150

Ans:-





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 :

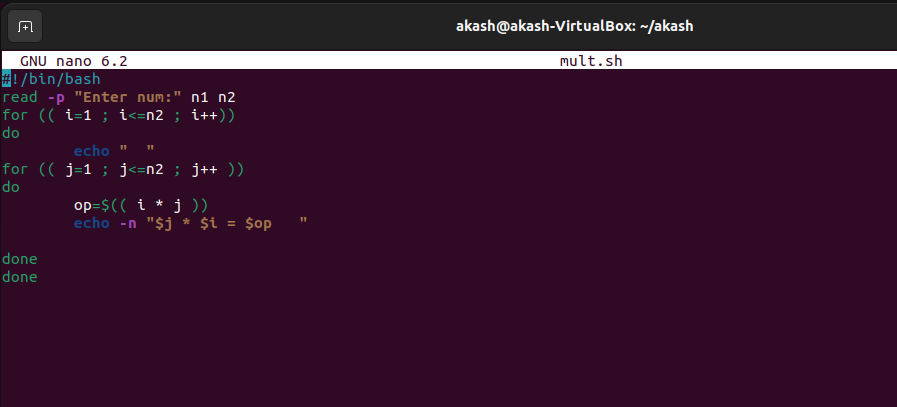
Multiplication table from 1 to 8

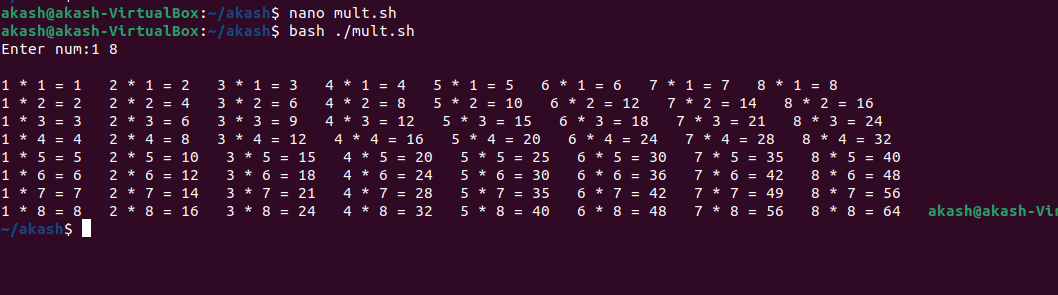
1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8

...

1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80

Ans:-





Q8. Write a Shell Script to display the n terms of odd natural numbers and their sum.

Test Data

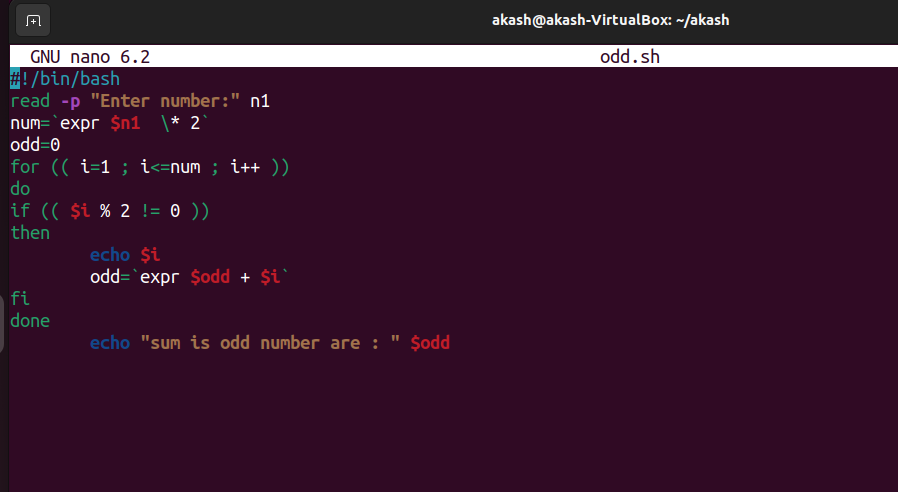
Input number of terms : 10

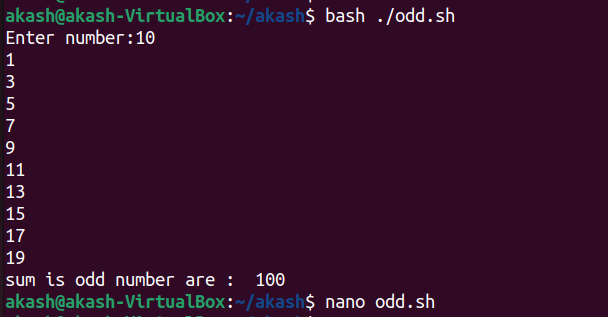
Expected Output :

The odd numbers are :1 3 5 7 9 11 13 15 17 19

The Sum of odd Natural Number upto 10 terms : 100

ANS :





Q9. Write a Shell Script to display a pattern like a right angle triangle using an asterisk.

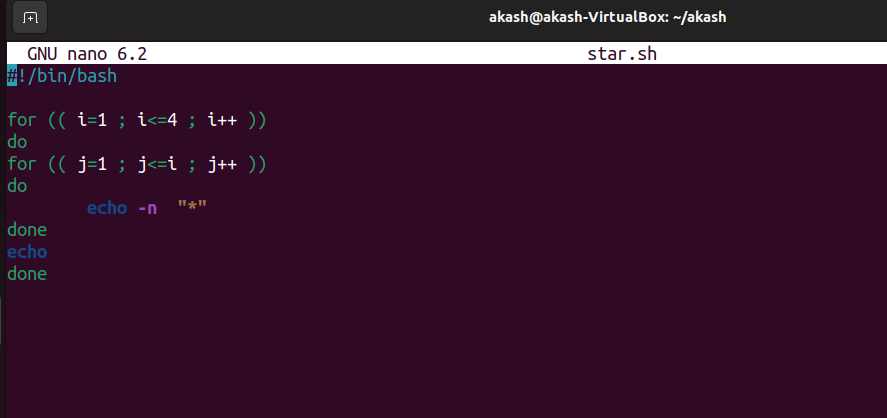
The pattern like :

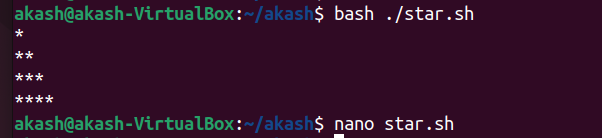
\*

\*\*

\*\*\*

\*\*\*\*





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

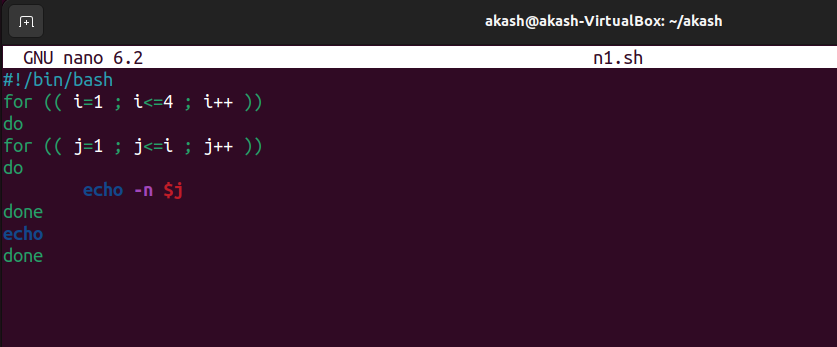
The pattern like :

1

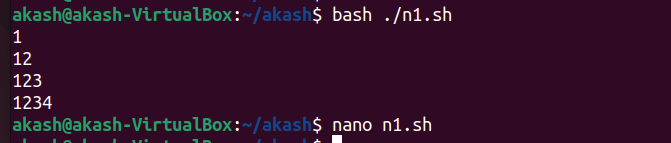
12

123

1234



ANS :



Q11. 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.

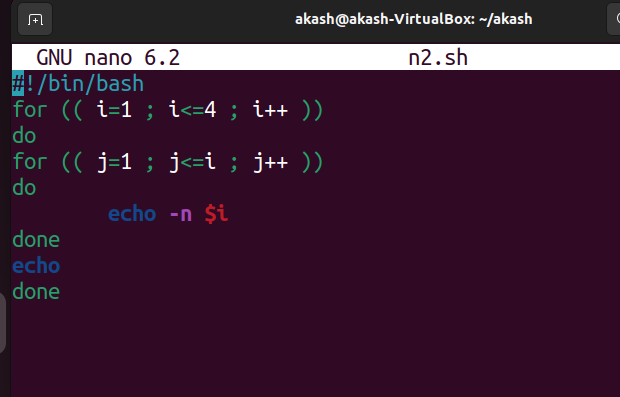
The pattern like :

1

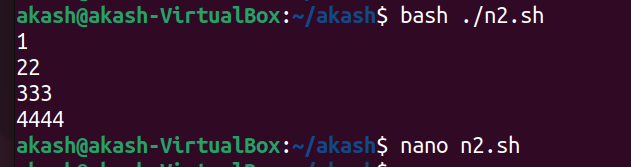
22

333

4444



ANS :



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

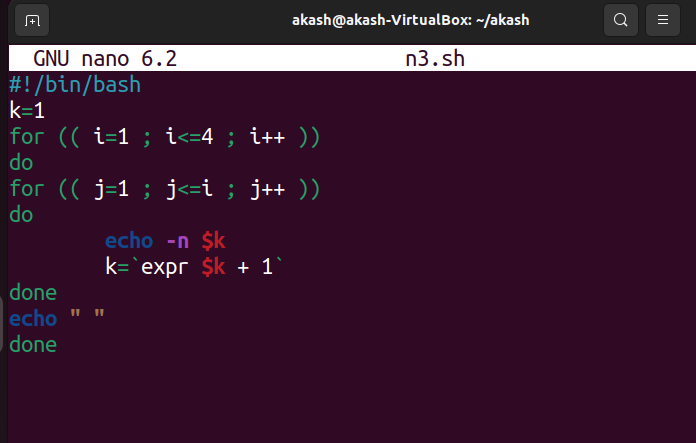
The pattern like :

1

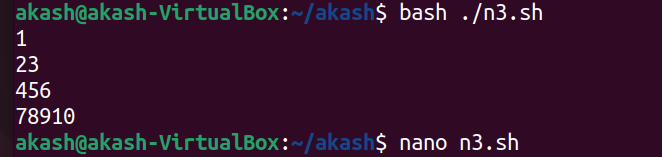
2 3

4 5 6

7 8 9 10



Ans :



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

1

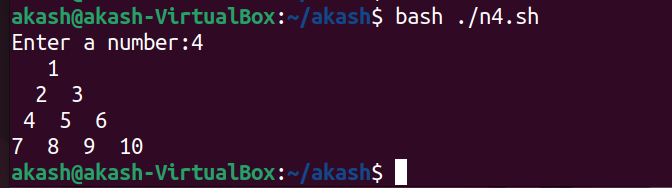
2 3

4 5 6

7 8 9 10



Ans:-



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

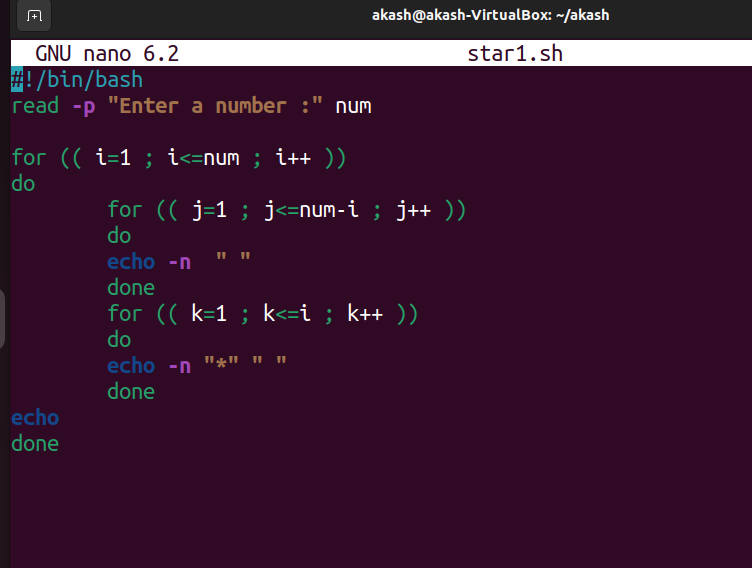
\*

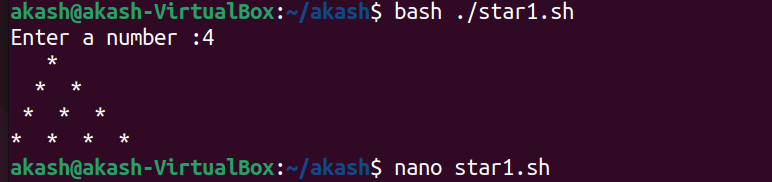
\* \*

\* \* \*

\* \* \* \*

Ans:-





15. Write a Shell Script to calculate the factorial of a given number.

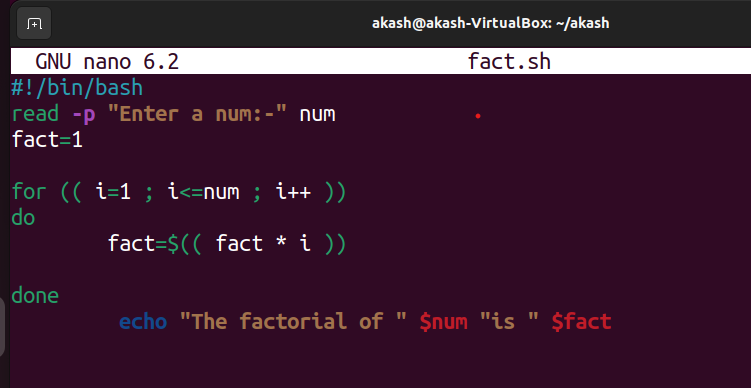
Test Data :

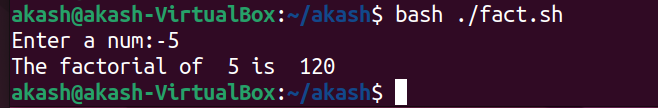
Input the number : 5

Expected Output :

The Factorial of 5 is: 120

Ans:-





16. Write a Shell Script to display the sum of n terms of even natural numbers.

Test Data :

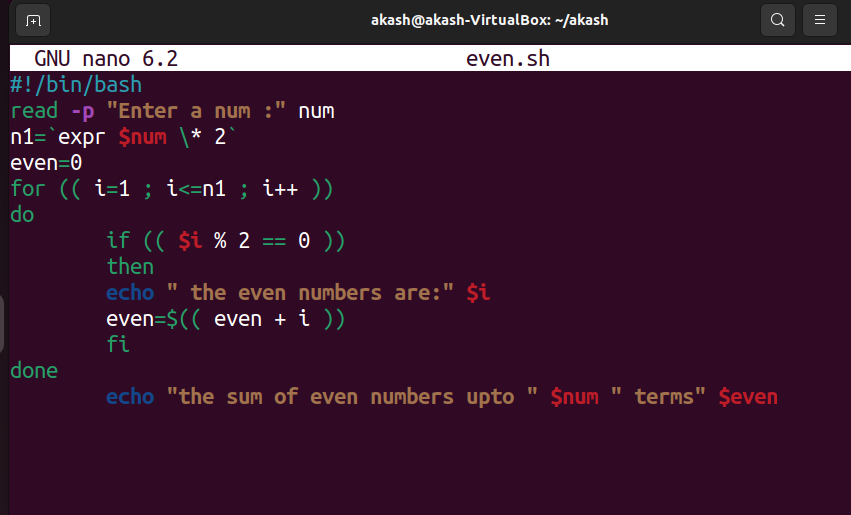
Input number of terms : 5

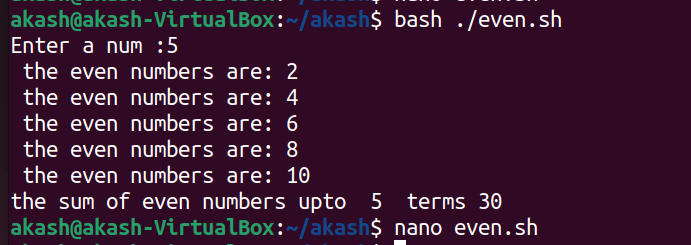
Expected Output :

The even numbers are :2 4 6 8 10

The Sum of even Natural Number upto 5 terms : 30

Ans:-





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.

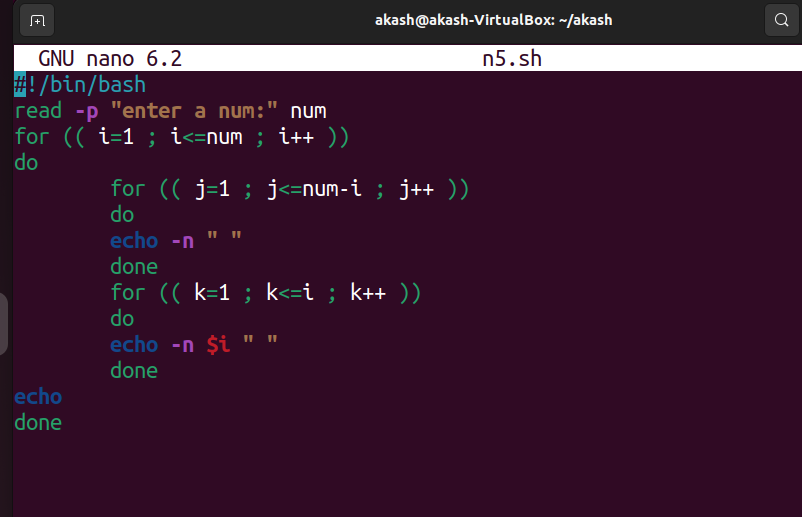
1

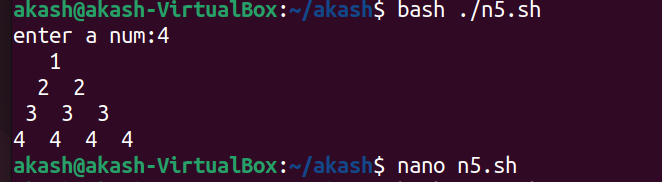
2 2

3 3 3

4 4 4 4

Ans:-





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

s=1.0 t=1.0

num=1.0 fac=1.0 y=2

m=1

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

for (( pr=1; pr<=y; pr++ )) do

fac=$(echo "$fac\*$pr" | bc -l) num=$(echo "$num\*$x" | bc -l)

done

m=$(echo "$m\*-1" | bc -l) num=$(echo "$num\*$m" | bc -l) t=$(echo "$num/$fac" | bc -l) s=$(echo "$s+$t" | bc -l) y=$(echo "$y+2" | bc -l) num=1.0

fac=1.0 done

echo "the sum = $s"

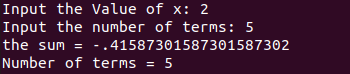
echo "Number of terms = $n" echo "value of x = $x"

Ans:-

Input the Value of x: 2 Input the number of terms: 5

the sum = -.41587301587301587302

Number of terms = 5 value of x = 2



**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=0.0

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

# print the terms of the series and update the sum for ((i=1; i<=n; i++))

do

term=$(echo "scale=6; 1.0 / $i" | bc) echo -n "1/$i"

if [ $i -ne $n ] then

echo -n " + "

fi

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

# print the sum of the series

echo -e "\nSum of Series upto $n terms : $sum"

Ans:-



**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

read -p "enter a number" n

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

for((j=0; j<n-i; j++)) do

echo -n " "

done

for((l=0; l<$(((2\*i)-1)); l++)) do

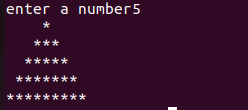
echo -n "\*"

done

echo -e " "

done

Ans:-



**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

read n term=0 sum=0

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

term=$((($term \* 10) + 9)) sum=$((term + sum))

echo -n "$term " done

echo $sum



**—---------------------------------------------------------------**

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

**1**

**01**

**101**

**0101**

**10101**

#!/bin/bash

read -p "Enter the number of rows: " n for ((i=1; i<=n; i++))

do

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

sum=$((i+j))

if [ $((sum % 2)) -eq 0 ]; then echo -n "1"

else

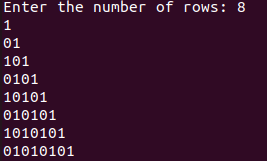
fi done

echo -n "0"

echo ""

done

Ans:-



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 : 16.37500**

#!/bin/bash

echo "Enter the value of x: "

read x

echo "Enter number of terms: " read n

sum=0 power=1 sign=1

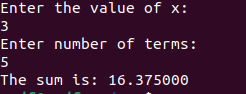
for ((i=1; i<=n; i++)) do

term=$(echo "scale=6; $sign \* $power" | bc -l) sum=$(echo "scale=6; $sum + $term" | bc -l) power=$(echo "scale=6; $power \* $x \* $x" | bc -l) sign=$(echo "scale=6; $sign \* -1" | bc -l)

done

echo "The sum is: $sum"

Ans:-



**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

echo "Enter the value of x:" read x

echo "Enter the number of terms:" read n

sum=0 sign=1 term=$x

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

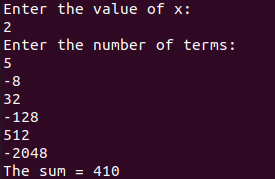
sum=$(echo "$sum + $sign\*$term" | bc) term=$(echo "$term \* $x \* $x" | bc) sign=$(echo "-1\*$sign" | bc)

echo $sum

done

echo "The sum = $sum"

Ans:-



**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 numbers upto 5 terms are :1 4 9 16 25 The Sum of Square Natural Number upto 5 terms = 55**

.

#square of number

read -p "Enter the Number of terms: " num

echo "The Square natural upto $num terms are: " for ((i=1; i<=num; i++))

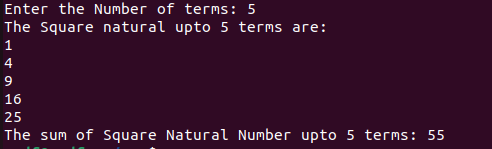
do

prod=$((i\*i)) echo $prod sum=$((sum+prod))

done

echo "The sum of Square Natural Number upto $num terms: "$sum

Ans:-



**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**

#1 + 11 + 111 + 1111 + ....

read -p "Enter the number of terms : " terms num=0

sum=0

for(( i=0; i<terms; i++ )) do

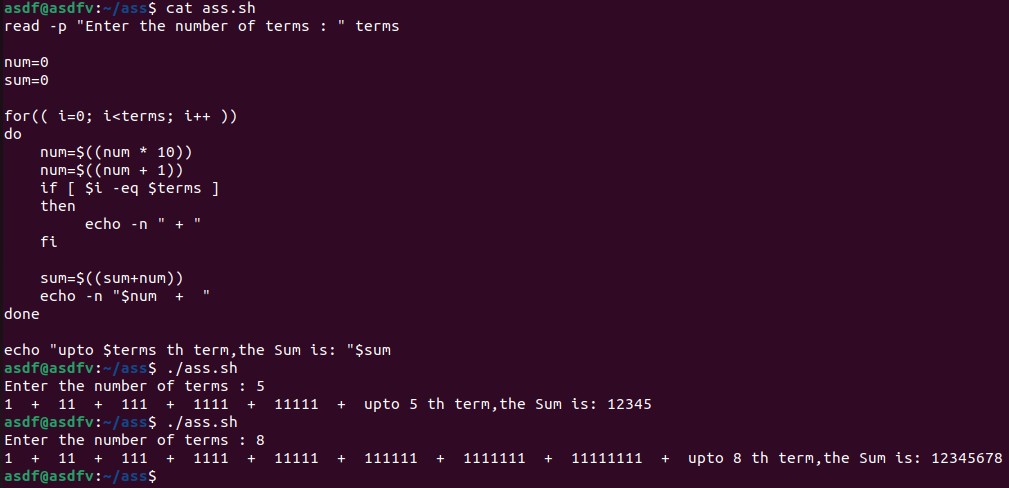
num=$((num \* 10))

num=$((num + 1))

if [ $i+1 -eq $terms ] then

echo -n " + " fi

sum=$((sum+num)) done

echo "The 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.**

#Perfect Number

read -p "Enter the number: " num echo "The Positive Divisor: " for(( i=1; i<num; i++ ))

do

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

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

fi done

echo "Sum of the Divisor is: " $sum if [ $sum -eq $num ]

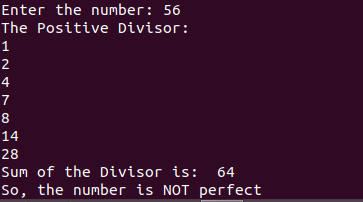
then

echo "So, the number is perfect" else

echo "So, the number is NOT perfect"

fi

Ans:-



**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 a range to find perfect numbers" n Echo “Perfect numbers in range are : “

perfect(){ divsum=0

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

rem=$(($1 % $i)) if [ $rem -eq 0 ] then

divsum=` expr $divsum + $i `

fi

done

if [ $divsum -eq $1 ] then

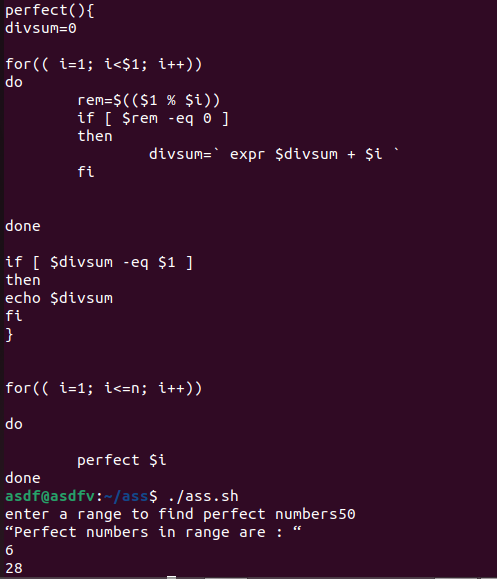
echo $divsum fi

}

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

perfect $i

done



**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 to check if its Armstrong: " n armstrng(){

x=$1 cubesum=0 rem=0

while [ $x -gt 0 ] do

rem=$(($x % 10))

cube=` expr $rem \\* $rem \\* $rem `

cubesum=` expr $cubesum + $cube ` x=$(($x / 10))

done

if [ $1 -eq $cubesum ] then

echo "$1 is an Armstrong Number"

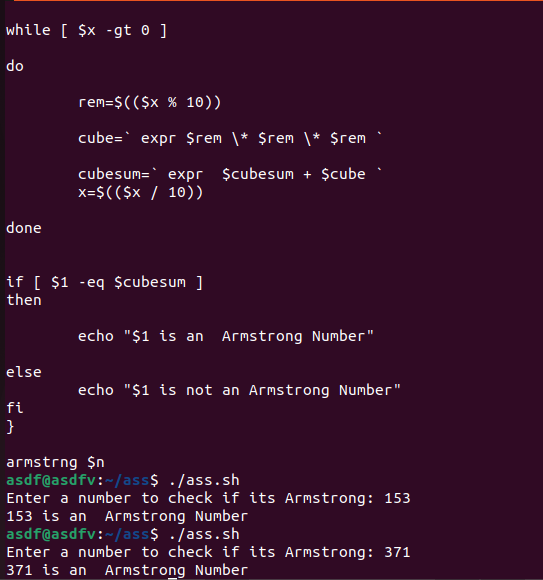
else

fi

}

echo "$1 is not an Armstrong Number"

armstrng $n



**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 range to print Armstrong numbers in the range: " n

armstrng(){ x=$1 cubesum=0 rem=0

while [ $x -gt 0 ] do

rem=$(($x % 10))

cube=` expr $rem \\* $rem \\* $rem `

cubesum=` expr $cubesum + $cube ` x=$(($x / 10))

done

if [ $1 -eq $cubesum ] then

echo -n "$1 "

fi

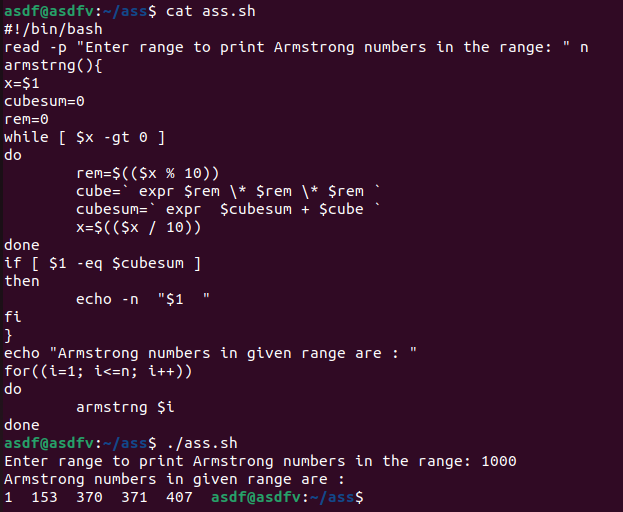
}

echo "Armstrong numbers in given range are : " for((i=1; i<=n; i++))

do

armstrng $i

done



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

**\***

**\*\*\***

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

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

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

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

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

**\*\*\***

**\***

.

#!/bin/bash

echo "Enter number of rows" read n

x=$((n-1))

for ((k=1;k<=n;k++)) do

for ((i=1;i<=x;i++)) do

printf " " done

x=$((x-1))

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

printf "\*" done

printf "\n"

done

x=1

for ((k=1;k<=n-1;k++)) do

for ((i=1;i<=x;i++)) do

printf " " done

x=$((x+1))

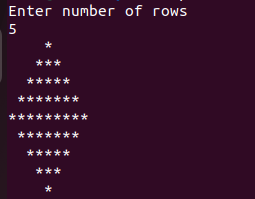
for ((i=1;i<=2\*(n-k)-1;i++)) do

printf "\*"

done

printf "\n"

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 a number: " num if [ $num -lt 2 ]; then

echo "$num is not a prime number" exit

fi flag=1

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

done

if [ $((num % i)) -eq 0 ]; then flag=0

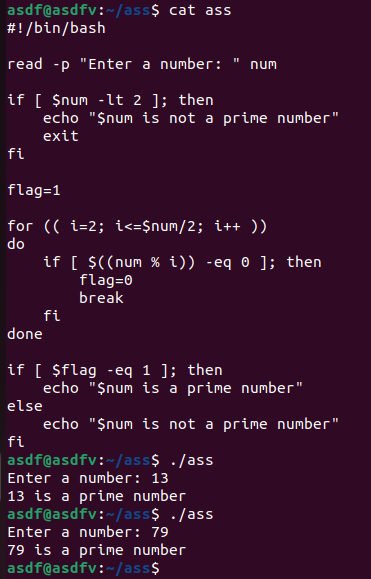
break fi

if [ $flag -eq 1 ]; then

echo "$num is a prime number"

else fi

echo "$num is not a prime number"



**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

echo "Enter the number of rows: " read rows

for ((i=0;i<rows;i++)) do

for ((j=0;j<rows-i-1;j++)) do

echo -n " " done

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

if [[ $j -eq 0 || $i -eq 0 ]]; then term=1

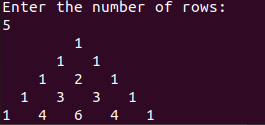
else

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

echo -n "$term " done

echo "" done

Ans:-



**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**

echo "Input starting number of range:"

read start

echo "Input ending number of range:" read end

echo "The prime numbers between $start and $end are:" for ((i=start; i<=end; i++))

do

flag=0

for ((j=2; j<i; j++)) do

if [ $((i%j)) -eq 0 ]; then flag=1

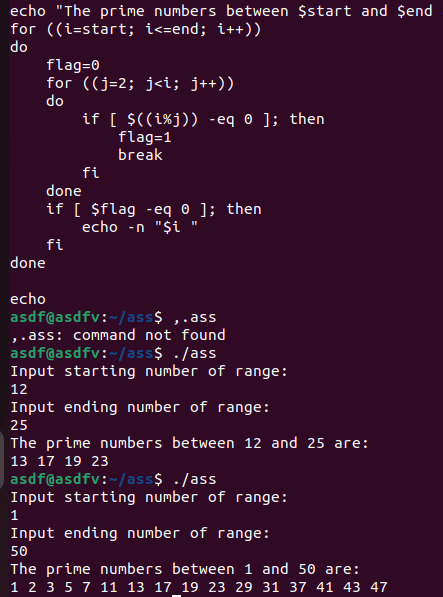
break

done echo

fi done

if [ $flag -eq 0 ]; then echo -n "$i "

fi



**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

echo "Enter the number of fibonacci terms tof display:" read n

a=0 b=1

echo "Here is the Fibonacci series up to $n terms:" echo -n "$a $b "

for (( i=3; i<=n; i++ )) do

c=$((a+b))

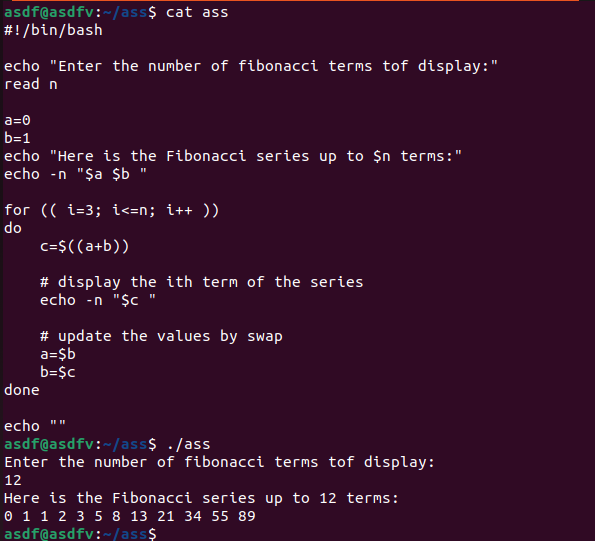
# display the ith term of the series echo -n "$c "

done

# update the values by swap a=$b

b=$c

echo ""



**36.Write a Shell Script to display 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

echo "Enter the number of rows:" read n

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

# print spaces

for (( j=i; j<n; j++ )) do

echo -n " " done

# print the first half of the row for (( j=1; j<=i; j++ ))

do

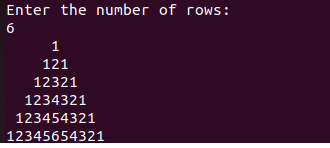
echo -n "$j" done

# print the second half of the row in reverse order for (( j=i-1; j>=1; j-- ))

do

echo -n "$j" done

echo "" done



1. **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

echo "Enter a number: " read num

rev=0

while [ $num -gt 0 ] do

lastdigit=$((num % 10)) rev=$((rev \* 10 + lastdigit))

num=$((num / 10)) done

echo "The number in reverse fashion is: $rev"

Ans:-



**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

reverse=0 n=$num

while [ $n -ne 0 ] do

done

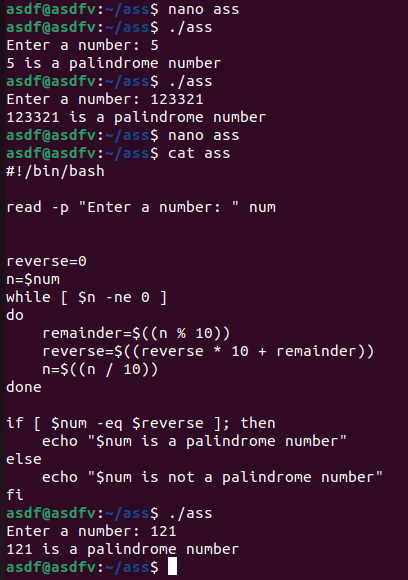
remainder=$((n % 10)) reverse=$((reverse \* 10 + remainder)) n=$((n / 10))

if [ $num -eq $reverse ]; then

echo "$num is a palindrome number"

else fi

echo "$num is not a palindrome number"



**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 sum=0

echo "Numbers between 100 and 200, divisible by 9:"

for ((i=100; i<=200; i++))

do

done

if [ $((i % 9)) -eq 0 ]; then echo -n "$i "

sum=$((sum + i)) fi

echo

echo "The sum: $sum"

Ans:-



**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 thr number of rows : " n lettr=65

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

for (( j=n; j>=i; j-- )) do

echo -n " " done

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

done

echo -n "$(printf \\$(printf '%03o' "$lettr"))" lettr=$((lettr+1))

lettr=$((lettr-1))

for (( m=1; m<i; m++ )) do

done

lettr=$((lettr-1))

echo -n "$(printf \\$(printf '%03o' "$lettr"))"

echo "" lettr=65

Done

Ans:-

