**OPERATING SYSTEM**

**LAB FILE**

**Abhijeet Singh & Kiran Fand**

Roll No. - 233002 & 233045

A. Create a basic calculator with using case.

#!/bin/bash

echo “This is a calculator”

echo :

echo 1. Addition

echo 2. Substraction

echo 3. Divsion

echo 4. Multiplication

read -p "Enter choice:" c

echo Choice $c selected:

case $c in

1)

read -p "Enter number 1:" one

read -p "Enter number 2:" two

echo Sum is:

expr $((one + two))

;;

2)

read -p "Enter number 1:" one

read -p "Enter number 2:" two

echo Difference is:

expr $((one - two))

;;

3)

read -p "Enter number 1:" one

read -p "Enter number 2:" two

echo Quotient is:

expr $((one / two))

;;

4)

read -p "Enter number 1:" one

read -p "Enter number 2:" two

echo Product is:

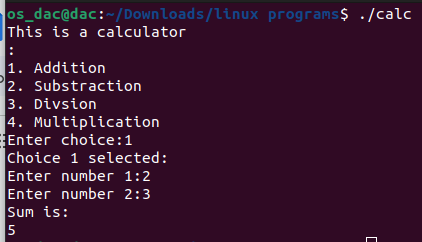
expr $((one \* two))

;;

\*)

echo Select Correct option

esac



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

#!/bin/bash

echo Greatest of 3 numbers

echo Enter number1

read num1

echo Enter number2

read num2

echo Enter number3

read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]

then

echo $num1 is greatest number

elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]

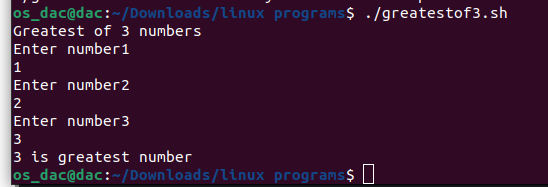
then

echo $num2 is greatest number

else

echo $num3 is greatest number

fi



C. 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" num

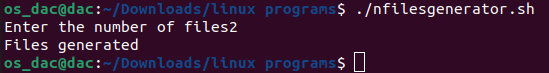
for ((i = 1; i<=num;i++));

do

touch file$i.txt

done

echo "Files generated"



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

#!/bin/bash

read -p "Enter a Number" num

if [ `expr $num % 2` -eq 0 ]

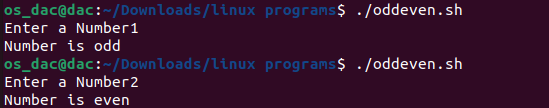
then

echo "Number is even"

else

echo "Number is odd"

Fi



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

#!/bin/bash

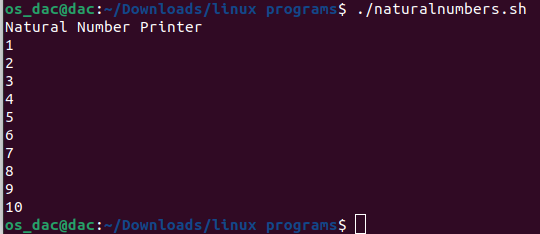
echo Natural Number Printer

for i in {1..10}

do

echo "$i"

done



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

#!/bin/bash

sum=0

i=1

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

do

sum=$((sum + i))

done

echo $sum



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

echo "Enter the value of n: "

read n

sum=0

echo "The first $n natural numbers are: "

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

do

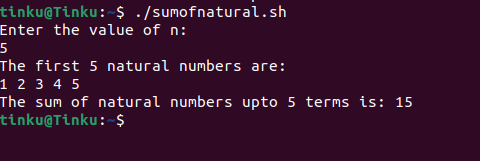
echo -n "$i "

sum=$((sum + i))

done

echo ""

echo "The sum of natural numbers upto $n terms is: $sum"



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

#!/bin/bash

sum=0

count=0

echo "Please enter 10 numbers:"

for i in {1..10}

do

read num

sum=$((sum+num))

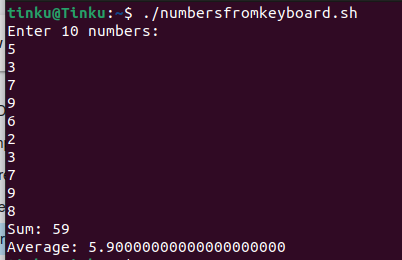
count=$((count+1))

done

avg=$(echo "scale=2; $sum / $count" | bc)

echo "The sum of the 10 numbers is: $sum"

echo "The average of the 10 numbers 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

echo "Enter the number of terms: "

read n

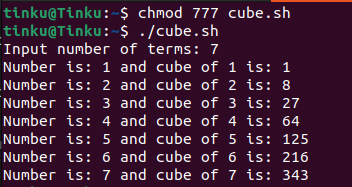
for ((i=1; i<=n; i++))

do

cube=$((i\*i\*i))

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

echo "Enter the number for the table: "

read n

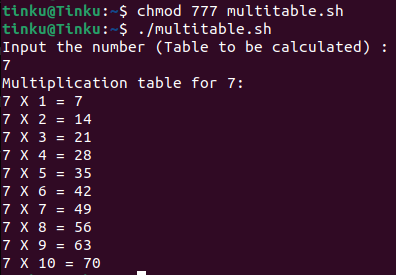
for ((i=1; i<=10; i++))

do

product=$((n\*i))

echo "$n X $i = $product"

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

echo "Enter the number for the table: "

read n

echo "Multiplication table from 1 to $n"

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

do

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

do

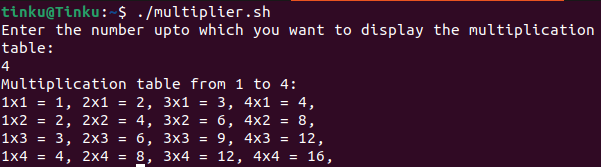
product=$((i\*j))

echo -n "$i x $j = $product, "

done

echo ""

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

echo "Input number of terms :"

read n

sum=0

echo -n "The odd numbers are: "

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

do

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

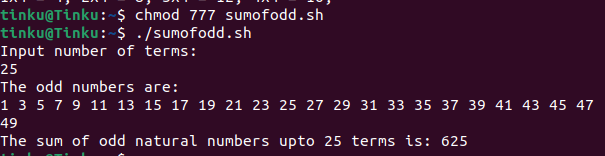
echo -n "$num "

sum=$((sum+num))

done

echo ""

echo "The Sum of odd Natural Number upto $n terms : $sum"



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

The pattern like :

\*

\*\*

\*\*\*

\*\*\*\*

#!/bin/bash

# This script displays a right angle triangle using asterisks

for (( i=1; i<=7; i++ ))

do

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

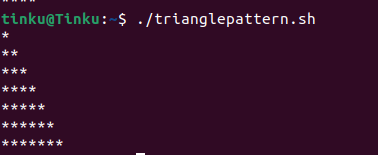
do

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

for (( i=1; i<=4; i++ ))

do

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

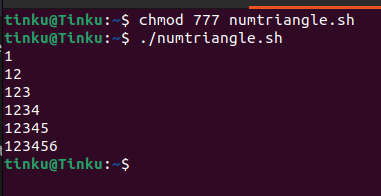
do

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

for (( i=1; i<=7; i++ ))

do

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

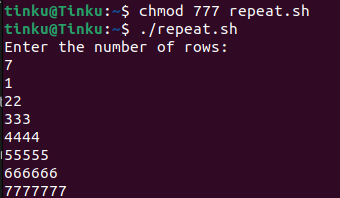
do

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

rows=4

num=1

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

do

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

do

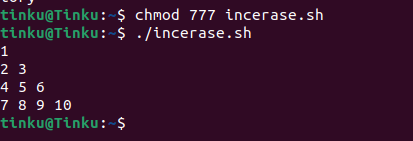
echo -n "$num "

num=$((num+1))

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

echo "Enter the number of rows:"

read rows

count=1

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

do

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

do

echo -ne " "

done

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

do

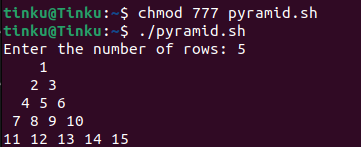
echo -ne "$count "

count=$((count+1))

done

echo ""

done



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

\*

\* \*

\* \* \*

\* \* \* \*

#!/bin/bash

echo "Enter the number of rows:"

read rows

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

do

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

do

echo -ne " "

done

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

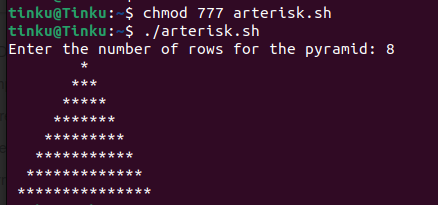
do

echo -ne "\*"

done

echo ""

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

echo "Enter a number:"

read num

fact=1

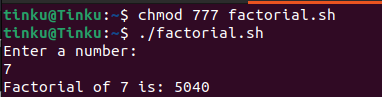
for (( i=2; i<=num; i++ ))

do

fact=$((fact \* i))

done

echo "Factorial of $num is: $fact"



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

echo "Enter the value of n: "

read n

sum=0

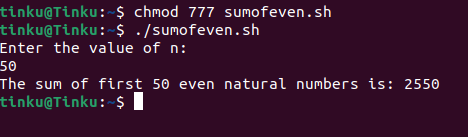
for (( i=2; i<=2\*n; i+=2 ))

do

sum=$(( sum+i ))

done

echo "The sum of first $n 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

echo "Enter the number of rows: "

read rows

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

do

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

do

echo -n " "

done

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

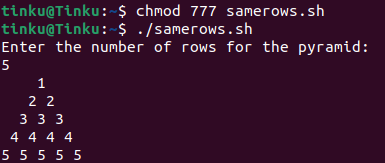
do

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

echo "Enter the value of x: "

read x

echo "Enter the number of terms: "

read n

sum=1

sign=-1

fact=2

power=2

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

do

sign=$(echo "$sign \* -1" | bc -l)

power=$(echo "$power \* 2" | bc -l)

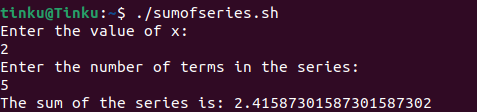
fact=$(echo "$fact \* ($(expr $i \\* 2 + 1)) \* ($(expr $i \\* 2 + 2))" | bc -l)

term=$(echo "$sign \* $x ^ $power / $fact" | bc -l)

sum=$(echo "$sum + $term" | bc -l)

done

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



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

echo "Enter the number of terms in the harmonic series: "

read n

sum=0

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

do

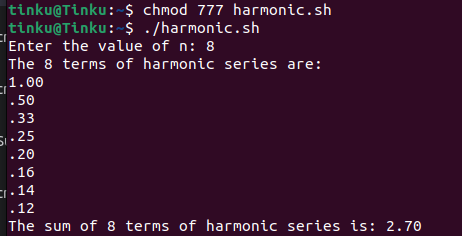
term=$(echo "scale=2; 1/$i" | bc)

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

echo "Term $i = $term"

done

echo "Sum of the harmonic series = $sum"



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

if [[ $# -eq 0 ]]; then

echo "Please provide the number of rows as an argument"

exit 1

fi

num\_rows=$1

midpoint=$((num\_rows/2 + 1))

for ((i=1; i<=num\_rows; i++)); do

spaces=$((midpoint-i))

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

printf " "

done

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

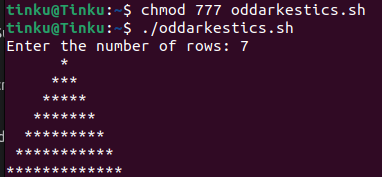
for ((j=1; j<=asterisks; j++)); do

printf "\*"

done

printf "\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

echo "Enter the number of terms in the series:"

read num\_terms

sum=0

for ((i=1; i<=num\_terms; i++)); do

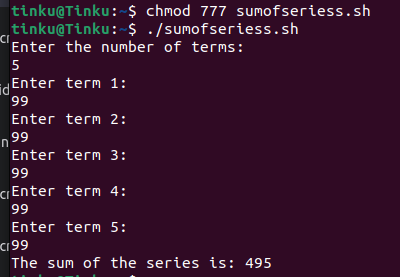
echo "Enter term $i:"

read term

sum=$((sum+term))

done

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



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

1

01

101

0101

10101

#!/bin/bash

echo "Enter the number of rows: "

read n

num=1

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

do

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

do

if [ $((i+j)) -eq 2 ] || [ $((i+j)) -eq 4 ]; then

echo -n "0"

else

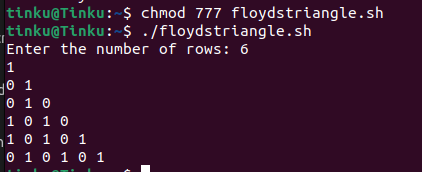
echo -n "1"

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

echo "Enter the value of x: "

read x

echo "Enter the number of terms: "

read n

sum=0

sign=1

power=1

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

do

term=$(echo "$x^$power" | bc)

term=$(echo "$term \* $sign" | bc)

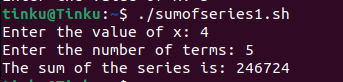
sum=$(echo "$sum + $term" | bc)

power=$((power+2))

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

done

echo "The sum of the series is: $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

echo "Input the value of x:"

read x

echo "Input number of terms:"

read n

sum=0

sign=1

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

do

term=$((x\*\*((2\*i)-1)))

term=$((term \* sign))

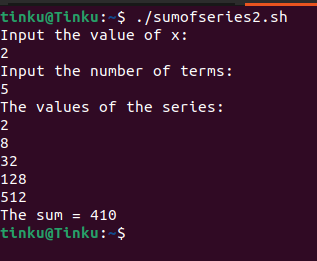
sign=$((sign \* -1))

sum=$((sum + term))

echo $term

done

echo "The sum = $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

echo "Input the number of terms:"

read n

sum=0

echo -n "The square natural upto $n terms are :"

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

do

term=$((i \* i))

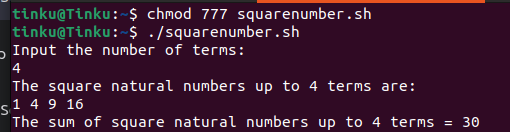
echo -n "$term "

sum=$((sum + term))

done

echo ""

echo "The Sum of Square Natural Number upto $n 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

echo "Input the number of terms:"

read n

sum=0

num=1

echo -n "The series is: "

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

do

echo -n "$num "

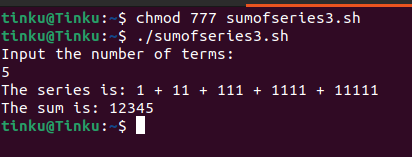
sum=$((sum + num))

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

done

echo ""

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.

#!/bin/bash

echo "Input the number:"

read n

sum=0

echo "The positive divisor :"

for ((i=1; i<=n/2; i++))

do

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

then

echo -n "$i "

sum=$((sum + i))

fi

done

echo ""

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

if [ $sum -eq $n ]

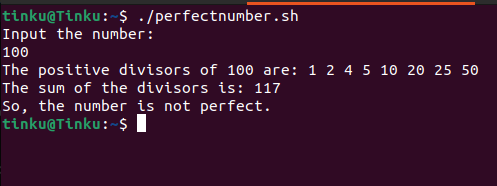
then

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 start

echo "Input the ending range of number:"

read end

echo "The Perfect numbers within the given range :"

for ((i=start; i<=end; i++))

do

sum=0

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

do

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

then

sum=$((sum + j))

fi

done

if [ $sum -eq $i ]

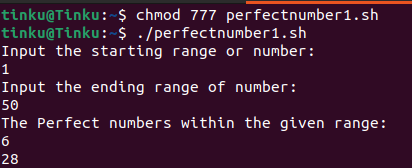
then

echo -n "$i "

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

echo "Input the number:"

read n

num\_digits=${#n}

sum=0

for ((i=0; i<num\_digits; i++))

do

digit=${n:$i:1}

sum=$((sum + digit \*\* num\_digits))

done

if [ $sum -eq $n ]

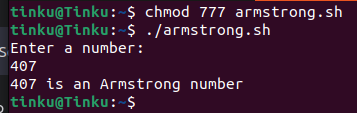
then

echo "The number $n is an Armstrong number."

else

echo "The number $n 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

# Script to find Armstrong numbers in a given range

echo "Enter the range: "

read range

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

do

sum=0

num=$i

while [ $num -gt 0 ]

do

digit=$(( $num % 10 ))

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

num=$(( $num / 10 ))

done

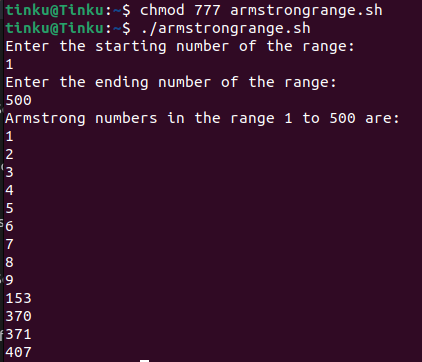
if [ $i -eq $sum ]

then

echo $i

fi

done



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

\*

\*\*\*

\*\*\*\*\*

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

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

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

\*\*\*\*\*

\*\*\*

\*

#!/bin/bash

echo "Enter the size of the diamond: "

read size

if (( $size % 2 == 0 )); then

echo "Size must be an odd number."

exit 1

fi

half\_size=$(( $size / 2 ))

spaces=$half\_size

asterisks=1

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

# Print spaces

for (( j=1; j<=$spaces; j++ )); do

echo -n " "

done

for (( j=1; j<=$asterisks; j++ )); do

echo -n "\*"

done

echo ""

if (( $i < $half\_size + 1 )); then

spaces=$(( $spaces - 1 ))

asterisks=$(( $asterisks + 2 ))

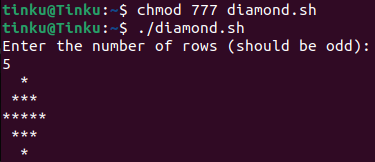
else

spaces=$(( $spaces + 1 ))

asterisks=$(( $asterisks - 2 ))

fi

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

echo "Enter a number:"

read n

if [ $n -lt 2 ]; then

echo "$n is not a prime number"

exit 0

fi

for ((i=2; i<$n; i++)); do

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

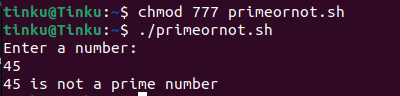
echo "$n is not a prime number"

exit 0

fi

done

echo "$n is 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 for Pascal's triangle:"

read rows

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

do

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

do

echo -n " "

done

coef=1

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

do

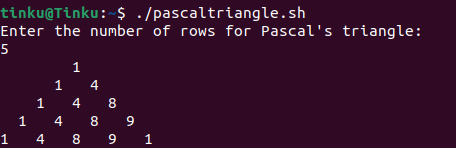
echo -n "$coef "

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

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

echo "Enter the lower limit of the range:"

read lower

echo "Enter the upper limit of the range:"

read upper

echo "Prime numbers between $lower and $upper are:"

for ((num=$lower; num<=$upper; num++))

do

prime=1

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

do

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

then

prime=0

break

fi

done

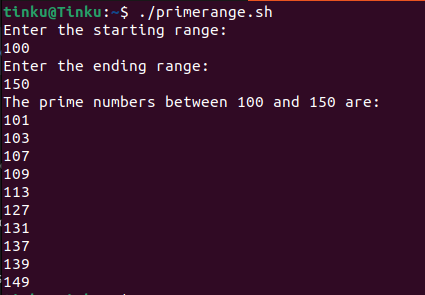
if [ $prime -eq 1 ] && [ $num -gt 1 ]

then

echo $num

fi

done



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 terms in the Fibonacci series:"

read n

# initialize first two terms

a=0

b=1

# print first n terms

echo "Fibonacci series:"

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

do

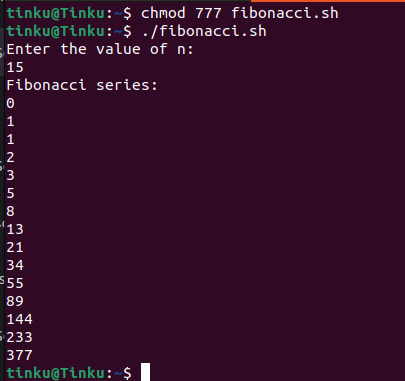
echo -n "$a "

next=$((a+b))

a=$b

b=$next

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

echo "Enter the number of rows:"

read n

echo "Pattern:"

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

do

# print leading spaces

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

do

echo -n " "

done

# print row

num=1

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

do

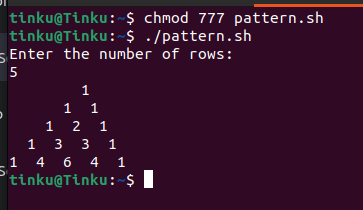
echo -n "$num "

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

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

echo "Enter a number:"

read num

reverse=""

while [ "$num" -gt 0 ]

do

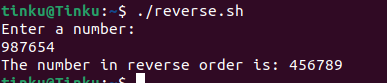
digit=$((num % 10))

reverse="${reverse}${digit}"

num=$((num / 10))

done

echo "The number in reverse order is: $reverse"



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

echo "Enter a number:"

read num

reverse=$(echo "$num" | rev)

if [ "$num" -eq "$reverse" ]

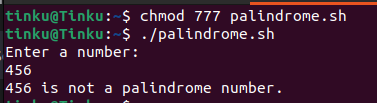
then

echo "$num is a palindrome number."

else

echo "$num is not a palindrome number."

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

sum=0

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

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

do

if [ $((i % 9)) -eq 0 ]

then

echo -n "$i "

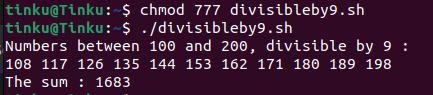
sum=$((sum + i))

fi

done

echo ""

echo "The sum: $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

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<=i; j++ ))

do

printf "%s " $(printf '\\x%02X ' $(($j+64)))

done

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

do

printf "%s " $(printf '\\x%02X ' $(($j+64)))

done

echo ""

done

