O.S. – Lab Assignment No.2 Shell Programming

Name : Rajnandini Nandkishor Dharashive

Class : SY CS-D

Roll No. : 5

PRN : 12211304

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Arithmetic Operations
2. Case Structure
3. If-else-then
4. Loops
5. CLA
6. functions
7. Array
8. String (length concatenation substring compare)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Arithmetic operations

Write a shell program to implement addition, subtraction, multiplication, division taking numbers as input from user.

1. Addition

Code:

#!/bin/bash

echo "Enter 1st number"

read n1

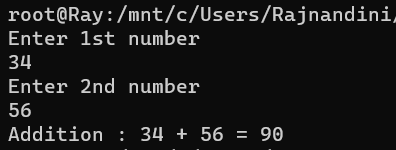
echo "Enter 2nd number"

read n2

add=$[$n1+$n2]

echo "Addition : $n1 + $n2 = $add"

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Subtraction

Code:

#!/bin/bash

echo "Enter 1st number"

read n1

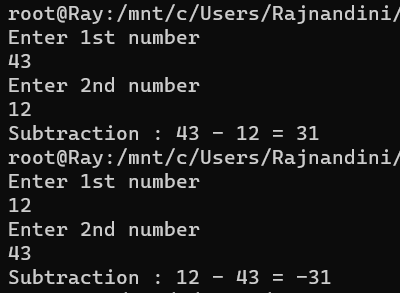
echo Enter 2nd number

read n2

sub=$(($n1-$n2))

echo Subtraction : $n1 - $n2 = $sub

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Multiplication

Code:

#!/bin/bash

echo Enter 1st number

read n1

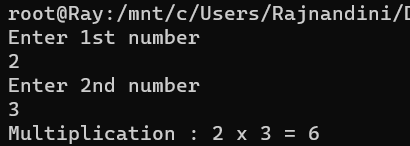
echo Enter 2nd number

read n2

mul=$[$n1\*$n2]

echo Multiplication : $n1 x $n2 = $mul

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Division

Code:

#!/bin/bash

echo "Enter 1st number"

read n1

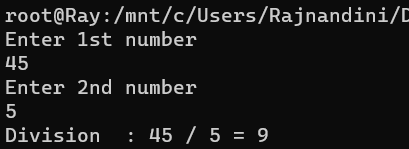
echo "Enter 2nd number"

read n2

div=$[$n1/$n2]

echo "Division (Int)  : $n1 / $n2 = $div"

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Arithmetic calculator (using case)

Write a shell program to implement calculator providing menu of different operations.

Code:

#!/bin/bash

echo "Enter 2 numbers"

read n1 n2

echo -e "\nMenu :\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n\nChoose operation : "

read choice

case $choice in

    1)

        echo -e "\nAddition    : $n1 + $n2 = $[$n1+$n2]"

    ;;

    2)

        echo -e "\nSubtraction : $n1 - $n2 = $[$n1-$n2]"

    ;;

    3)

        echo -e "\nMultiply    : $n1 x $n2 = $[$n1\*$n2]"

    ;;

    4)

        echo -e "\nDivision    : $n1 / $n2 = $[$n1/$n2]"

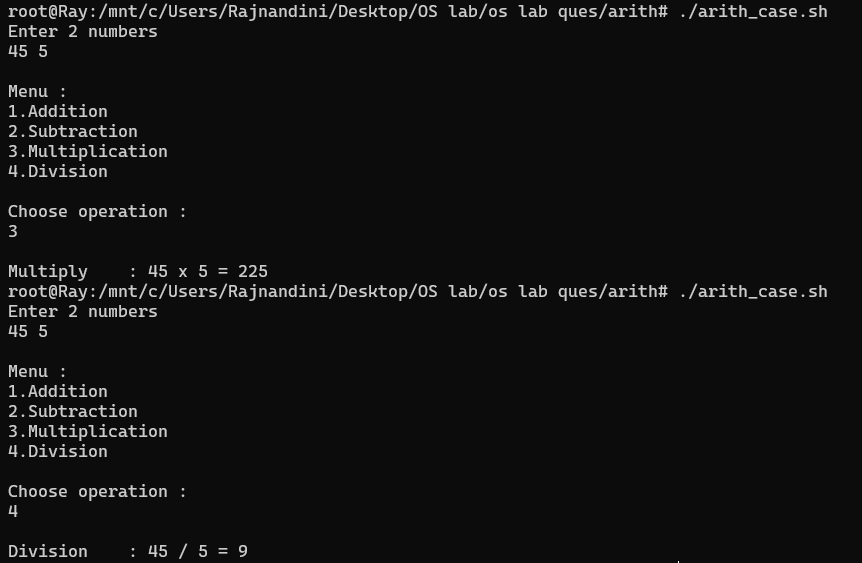
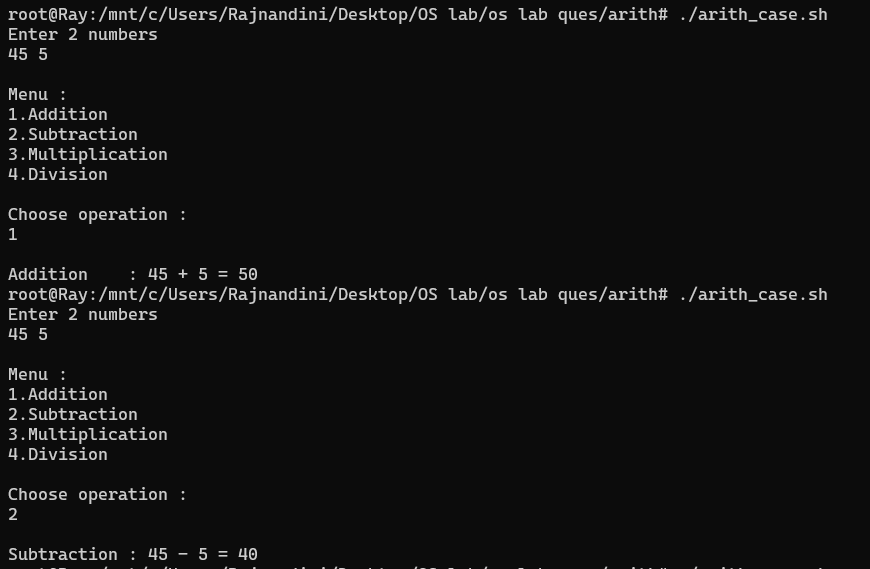
    ;;

\*)

;;

esac

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Case (using range)

Write a shell program to display if an input character is lowercase, uppercase, digit or special character.

Code:

#!/bin/bash

echo "Enter a character"

read char

case $char in

    \*[A-Z]\* )

            echo -e "It's an uppercase letter\n"

        ;;

    \*[a-z]\* )

            echo -e "It's a lowercase letter\n"

        ;;

    \*[0-9]\* )

            echo -e "It's a digit\n"

        ;;

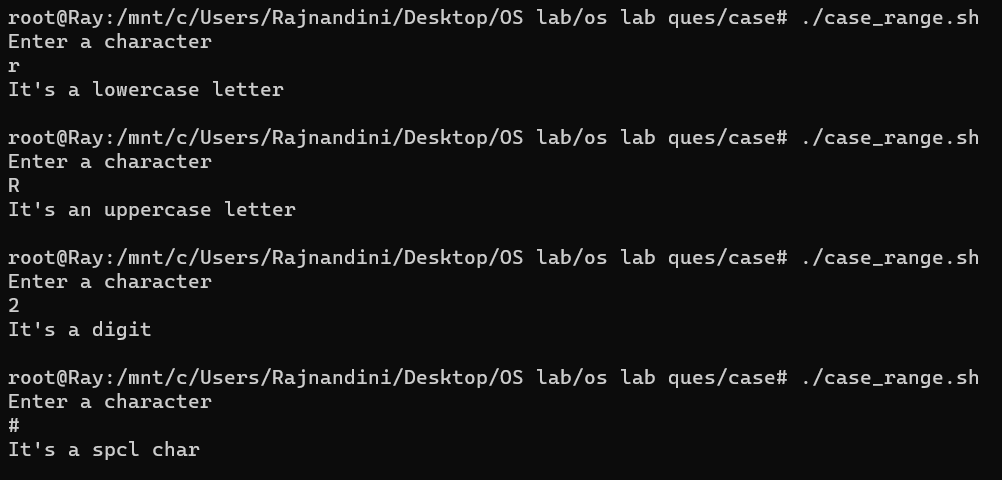
    \*)

            echo -e "It's a spcl char\n"

        ;;

esac

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. If then else

Write a shell program to compare two user input numbers.

Code:

#!/bin/bash

echo "\*\*\*  Program to compare 2 numbers  \*\*\*"

echo "Enter the 2 numbers"

read n1 n2

if [ $n1 -gt $n2 ]

    then

        echo "$n1 > $n2"

elif [ $n1 -lt $n2 ]

    then

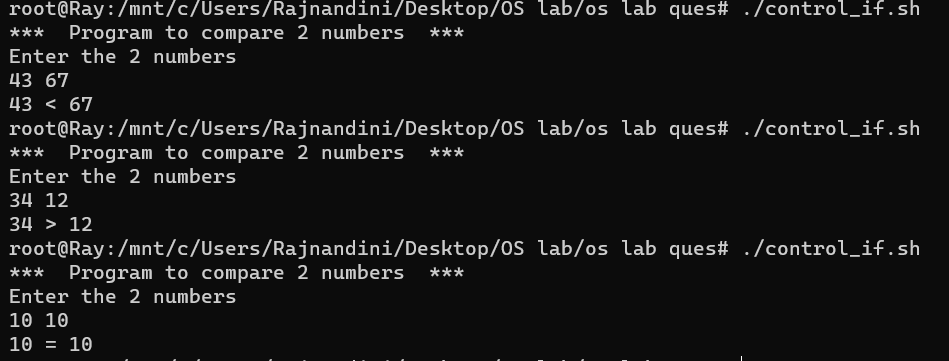
        echo "$n1 < $n2"

else

        echo "$n1 = $n2"

fi

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **LOOPS** :
2. For Loops:
3. Simple

Write a shell program to table of a number using simple for loop.

Code:

#!/bin/bash

echo Enter number

read num

echo Table of $num

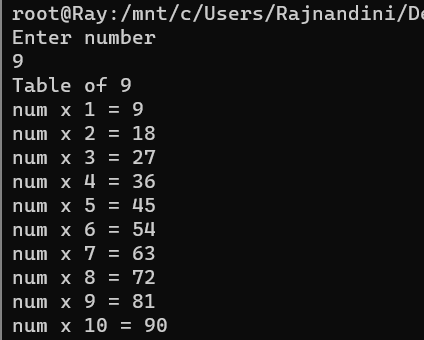
for n in 1 2 3 4 5 6 7 8 9 10;

    do

        echo num x $n = $[$n \* $num]

    done

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Range-based

Write a shell program to display table of user input number using range-based for loop.

Code:

#!/bin/bash

echo "Enter number"

read num

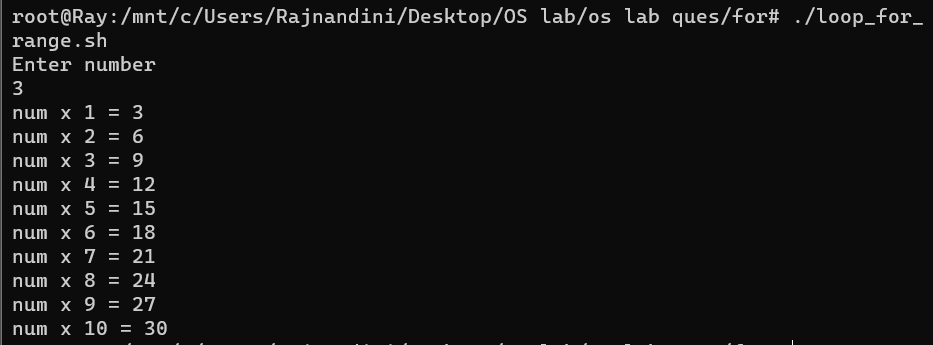
for n in {1..10};

    do

        echo num x $n = $[$n \* $num]

    done

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Array iteration

Write a shell program to implement iteration through a predefined array.

Code:

#!/bin/bash

echo ARRAY USING FOR LOOP

arr1=("a" "b" "c" "d" "e")

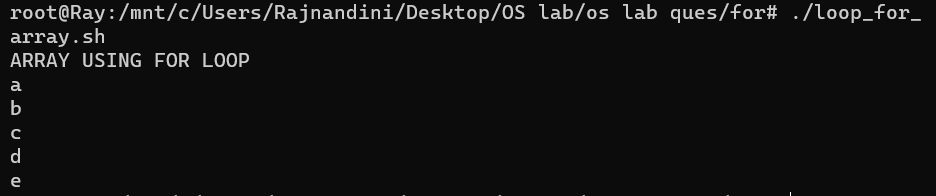
for n in ${arr1[@]};

    do

        echo $n

    done

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. C-style

Write a shell program to print even numbers from 1-10 using C-styled for loop.

Code:

#!/bin/bash

echo PRINT EVEN NUMBERS FROM 1-10 USING C-STYLED FOR LOOP

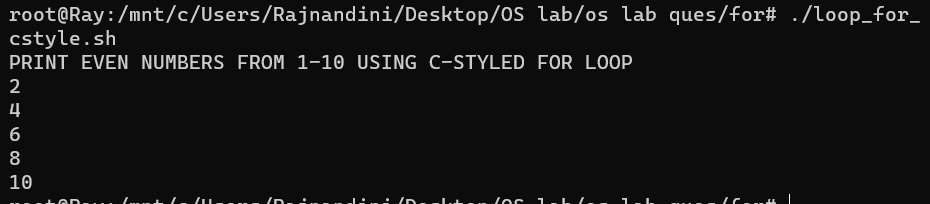
for((i=2;i<=10;i+=2));

    do

        echo $i

    done

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Infinite

Write a shell program to print even numbers from 10-20 using infinite for loop.

Code:

#!/bin/bash

echo PRINT EVEN NUMBERS FROM 10-20 USING INFINITE LOOP

n=10

for(( ; ; ));

    do

        if [ $n -gt 20 ]

            then

                break

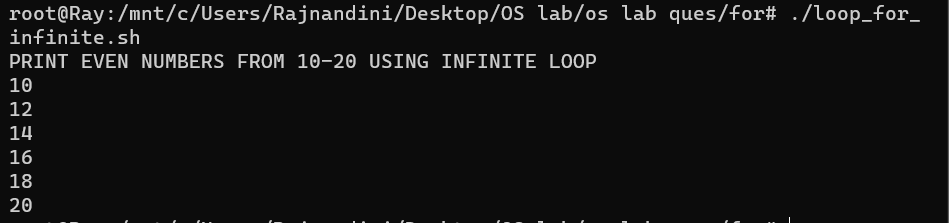
        fi

        echo $n

        ((n=$n+2))

    done

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Until loop

Write a shell program to print integers from 1-N using until loop taking N as input from user.

Code:

#!/bin/bash

echo PRINT INTEGERS FROM 1-N USING UNTIL LOOP

a=1

echo Enter number

read N

echo

until [ $a -eq $N ]

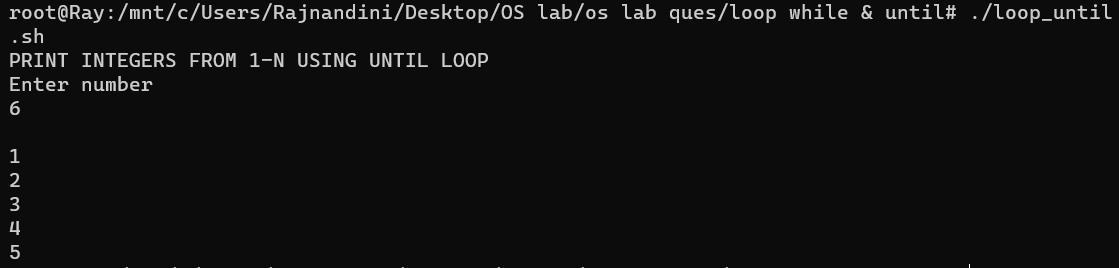
    do

        echo $a

        ((a=$a+1))

    done

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. While loop

Write a shell program to print first 10 integers and their squares using while loop.

Code:

#!/bin/bash

echo "PRINT 1ST 10 INTEGERS & THEIR SQUARES USING WHILE LOOP"

n=1

while [ $n -le 10 ]

do

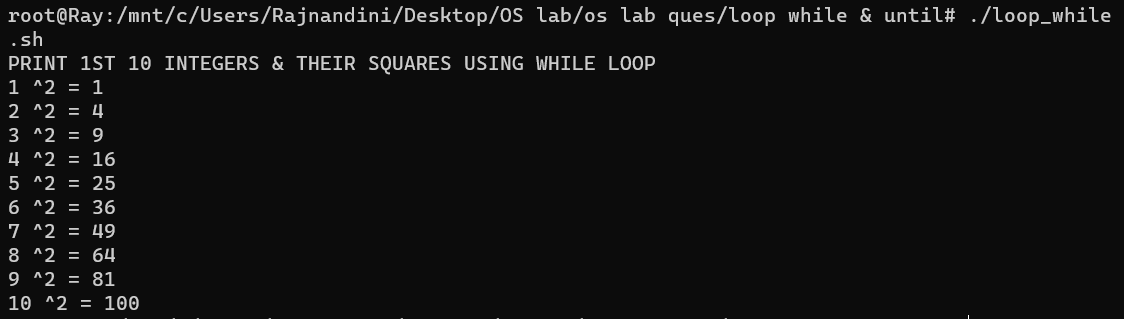
    ((sq=$n\*$n))

    echo "$n ^2 = $sq"

    ((n=$n+1))

done

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. CLA

Write a shell program to find average of three numbers using command line argument.

Code:

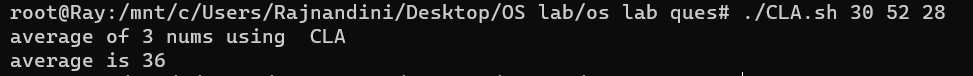
#!/bin/bash

echo "average of 3 nums using  CLA"

((avg=($1+$2+$3)/3))

echo "average is $avg"

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Function

Write a shell program to print table of input number using function.

Code:

#!/bin/bash

echo FUNCTION to print table of input number

table(){

    echo Table of $num

    for i in {1..10}

        do

            echo $num x $i = $[$num \* $i]

        done

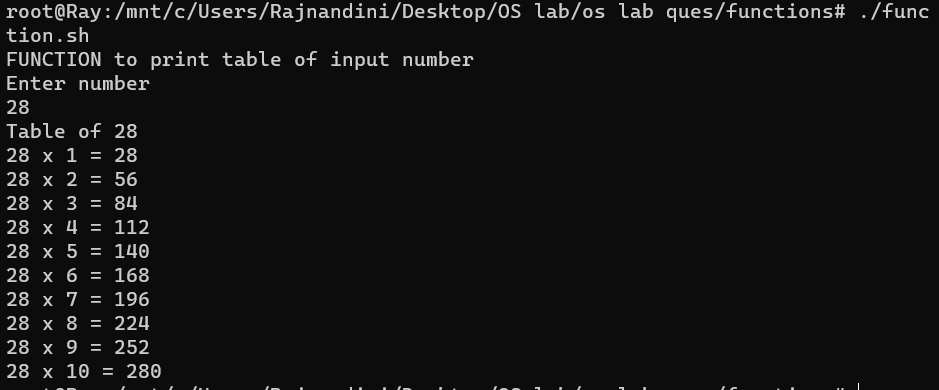
}

echo Enter number

read num

table

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Array
2. Simple Array

Write a shell program to display elements in an array

Code:

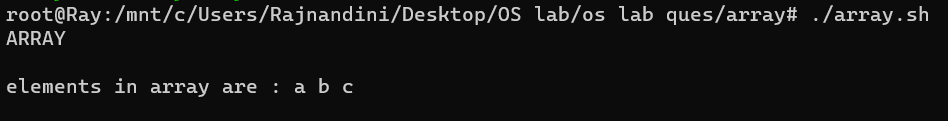
#!/bin/bash

echo -e "ARRAY\n"

arr1=("a" "b" "c")

echo -e "elements in array are : ${arr1[@]}\n"

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Check city in array

Write a shell program to find out whether user input city is present in a pre-defined array of cities or not.

Code:

#!/bin/bash

arr2=("Mumbai" "Pune" "Nagpur" "Delhi")

echo -e "\nEnter city :"

read city

count=0

for n in ${arr2[@]};

do

    if [ $city == $n ]

        then

            echo -e "\nCity found\n"

            break

    else

        ((count=$count+1))

    fi

done

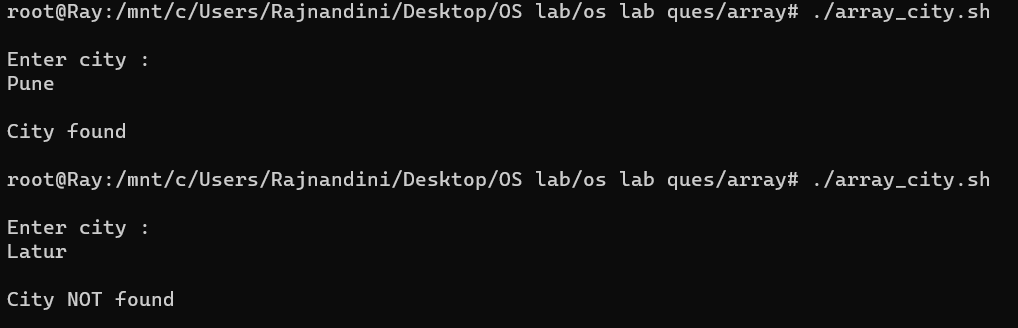
if [ $count -eq 4 ]

then

    echo -e "\nCity NOT found\n"

fi

Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. String (length concatenation substring compare)

Write a shell program to perform string operations using a menu-based approach for finding length of string, concatenation of two strings, finding substring of a string, and comparing two strings.

Code:

#!/bin/bash

echo "STRING operations"

echo "Enter string"

read str

echo -e "\nMenu :\n1.length\n2.concat 2 strings\n3.compare 2 strings\n4.substring\n\nChoose operation : "

read choice

case $choice in

    1)

        echo "length of $str is : ${#str}"

    ;;

    2)

        echo "Enter 2nd string"

        read str2

        echo "concating $str with $str2 : $str$str2"

    ;;

    3)

        echo "Enter 2nd string"

        read str2

        echo "Comparing strings :"

        if [ "$str" == "$str2" ]

        then

            echo "strings are same"

        else

            echo "strings are NOT same"

        fi

    ;;

    4)

        echo "Enter starting & ending index of substring : "

        read n1 n2

        echo -e "\nsubstring from $n1 to $n2 is    : ${str:$n1:$n2}"

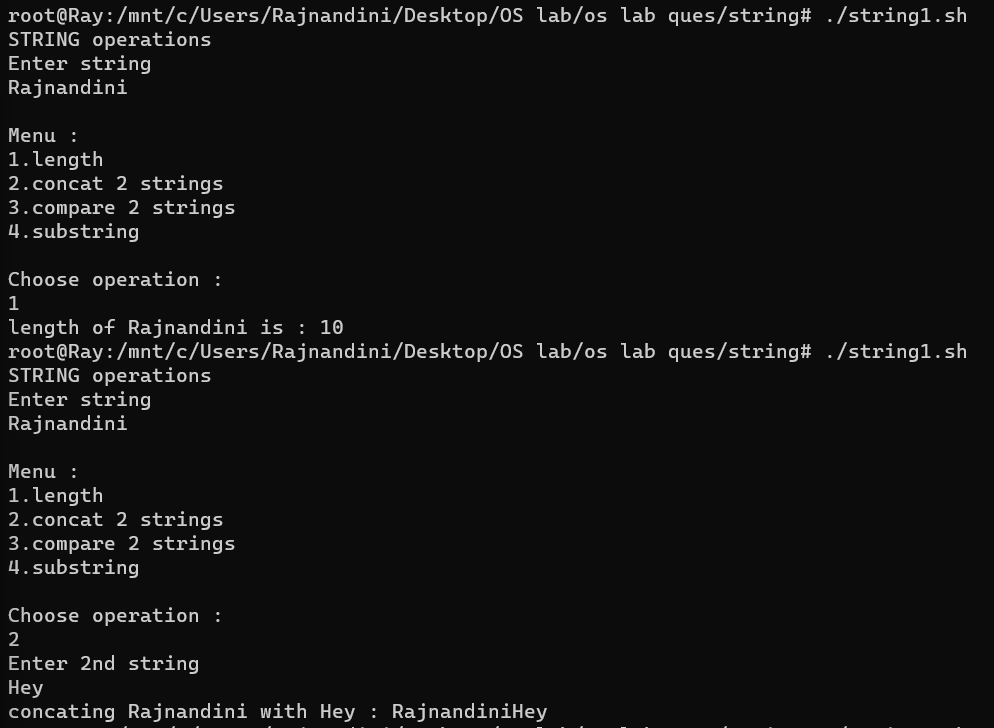
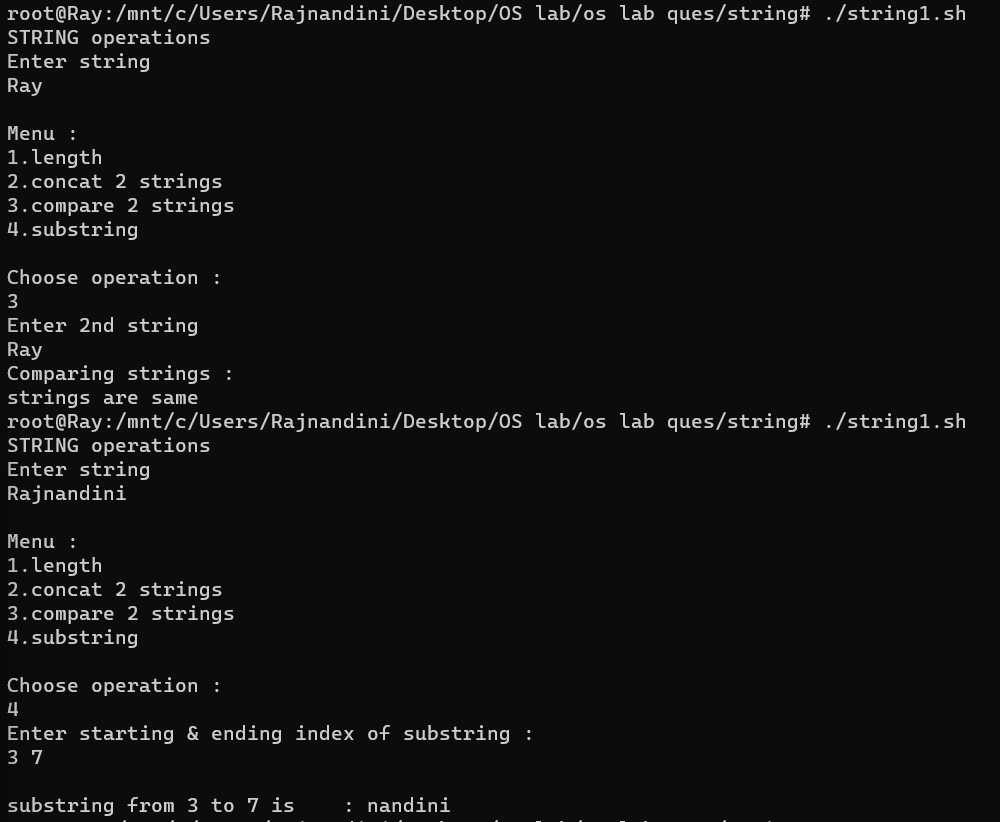
    ;;

\*)

;;

esac

Output:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_