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

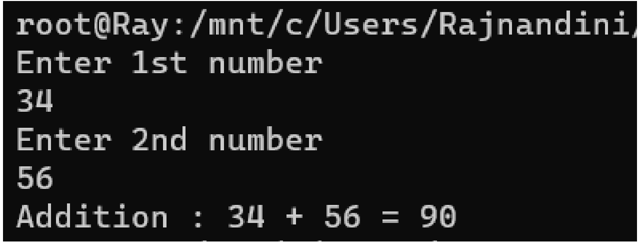
Write a shell program to implement addition, subtraction, multiplication, division taking numbers as input from user.

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:



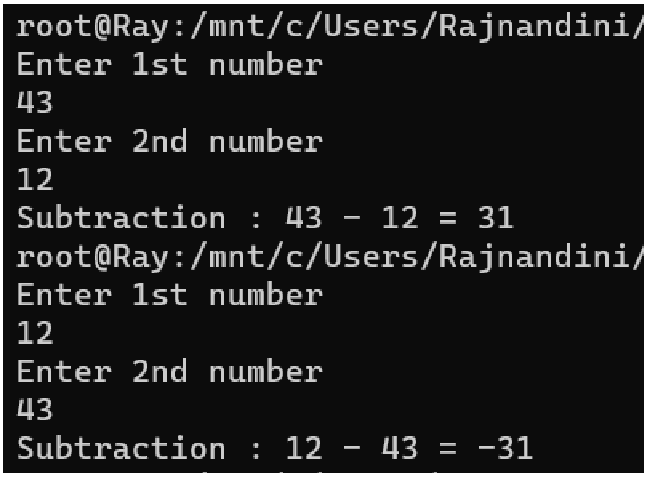
```
root@Ray:/mnt/c/Users/Rajnandini/
Enter 1st number
34
Enter 2nd number
56
Addition : 34 + 56 = 90
```

2. 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:



```
root@Ray:/mnt/c/Users/Rajnandini/
Enter 1st number
43
Enter 2nd number
12
Subtraction : 43 - 12 = 31
root@Ray:/mnt/c/Users/Rajnandini/
Enter 1st number
12
Enter 2nd number
43
Subtraction : 12 - 43 = -31
```

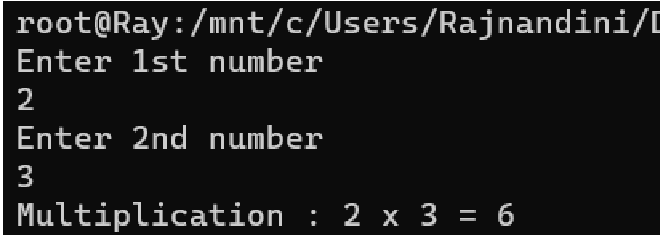
---

3. 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:



```
root@Ray:/mnt/c/Users/Rajnandini/D
Enter 1st number
2
Enter 2nd number
3
Multiplication : 2 x 3 = 6
```

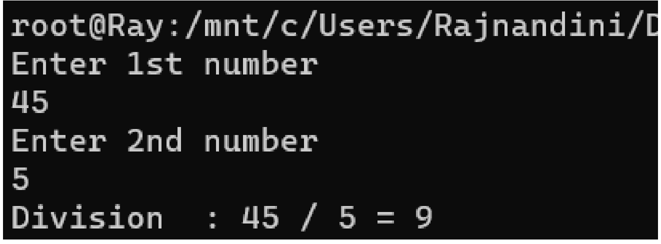
---

4. 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:



```
root@Ray:/mnt/c/Users/Rajnandini/D
Enter 1st number
45
Enter 2nd number
5
Division   : 45 / 5 = 9
```

---

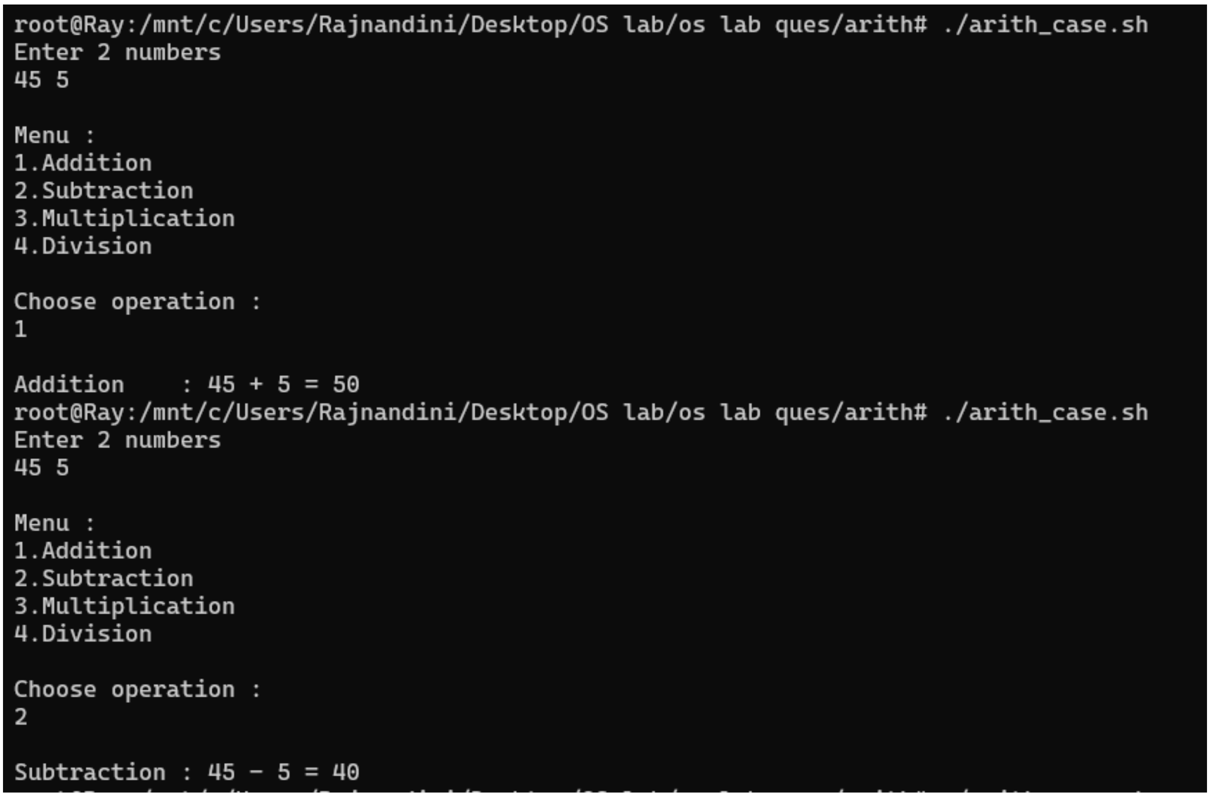
5. <u>Arithmetic calculator (using case)</u>

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

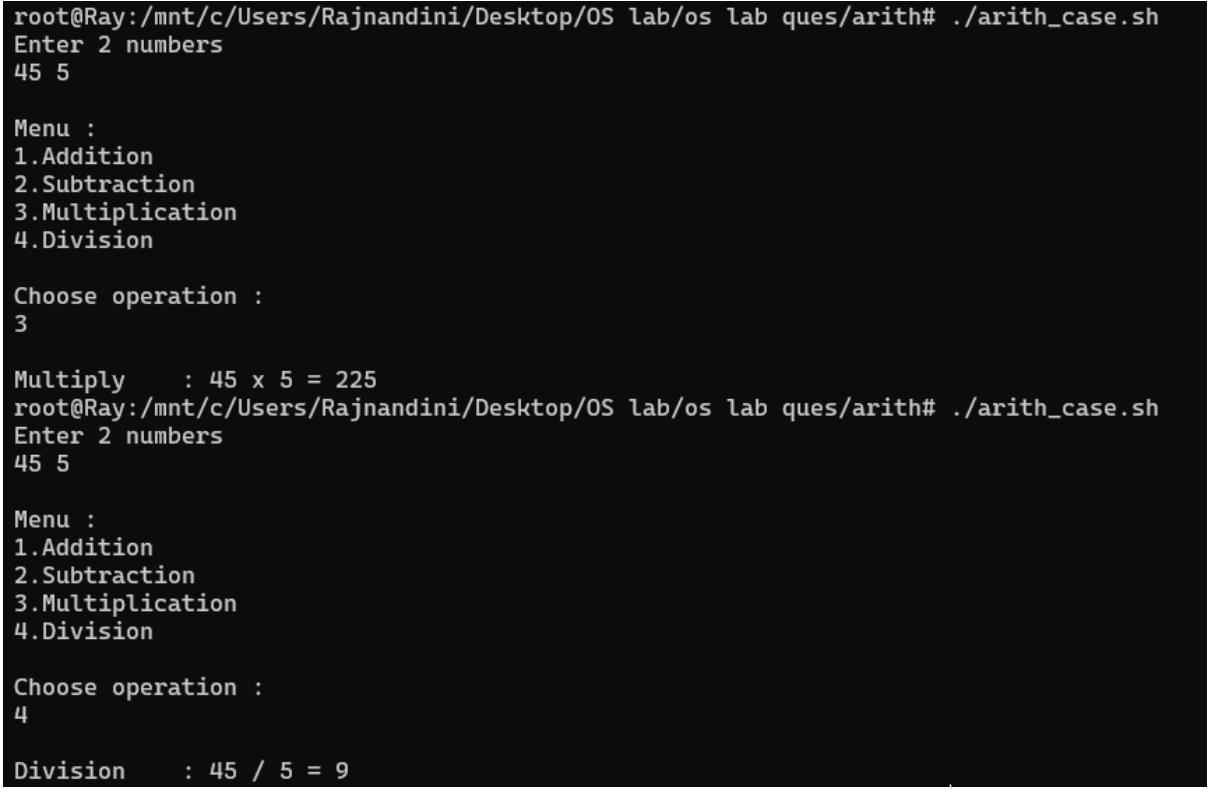
   Code:

```
#!/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:



```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/arith# ./arith_case.sh
Enter 2 numbers
45 5

Menu :
1.Addition
2.Subtraction
3.Multiplication
4.Division

Choose operation :
1

Addition    : 45 + 5 = 50
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/arith# ./arith_case.sh
Enter 2 numbers
45 5

Menu :
1.Addition
2.Subtraction
3.Multiplication
4.Division

Choose operation :
2

Subtraction : 45 - 5 = 40
```

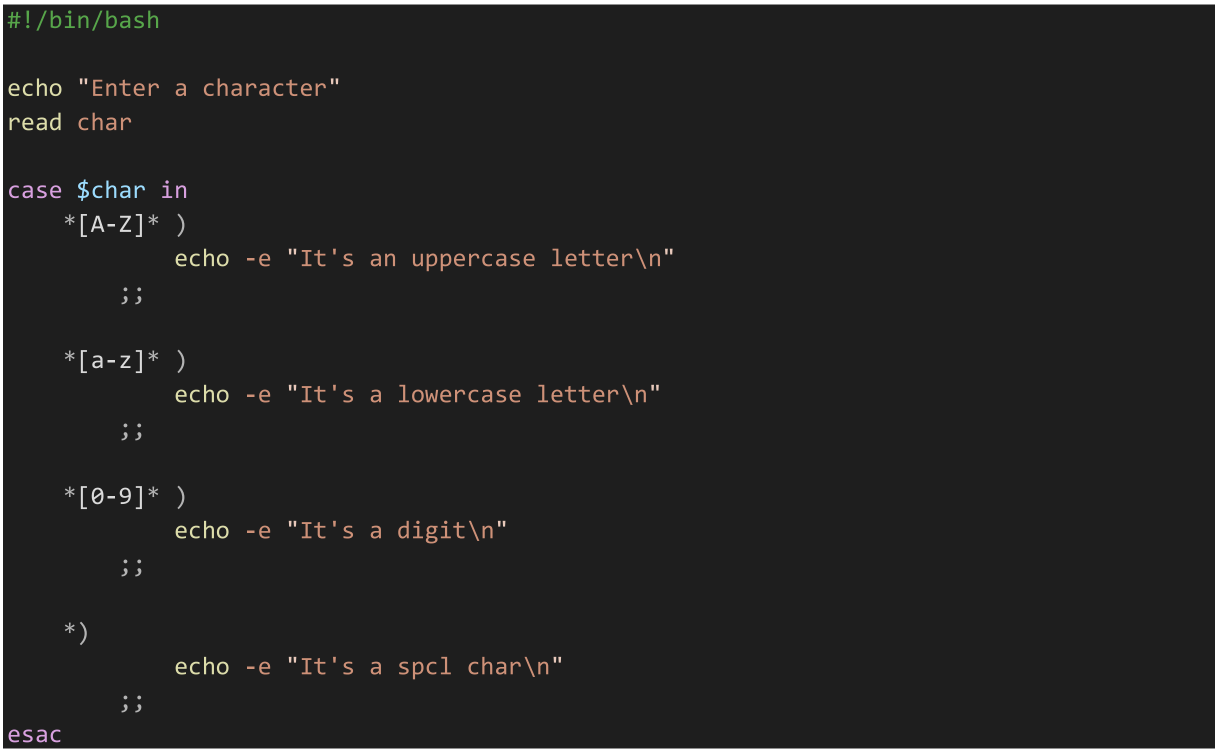


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/arith# ./arith_case.sh
Enter 2 numbers
45 5

Menu :
1.Addition
2.Subtraction
3.Multiplication
4.Division

Choose operation :
3

Multiply    : 45 x 5 = 225
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/arith# ./arith_case.sh
Enter 2 numbers
45 5

Menu :
1.Addition
2.Subtraction
3.Multiplication
4.Division

Choose operation :
4

Division    : 45 / 5 = 9
```

2) Case (using range)

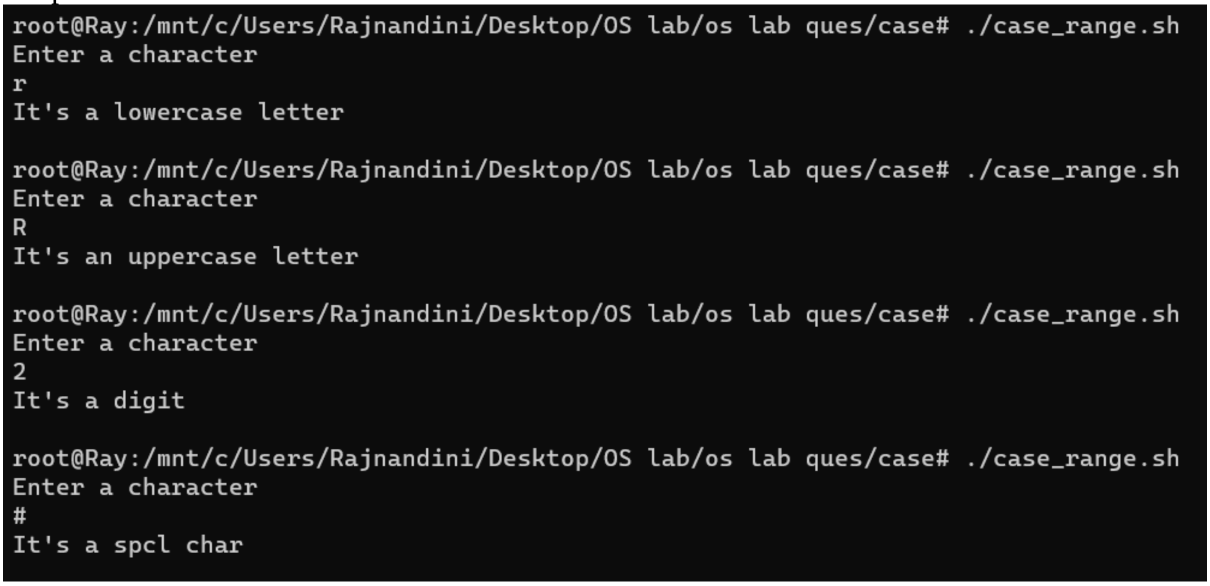
Write a shell program to display if an input character is lowercase, uppercase, digit or special character.

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:

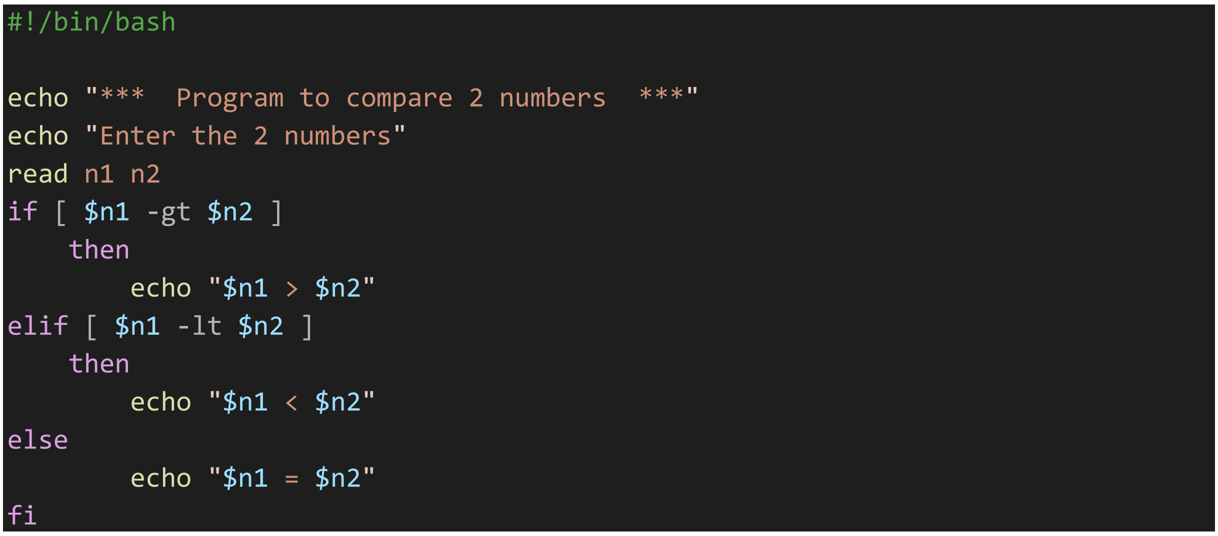


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/case# ./case_range.sh
Enter a character
r
It's a lowercase letter

root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/case# ./case_range.sh
Enter a character
R
It's an uppercase letter

root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/case# ./case_range.sh
Enter a character
2
It's a digit

root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/case# ./case_range.sh
Enter a character
#
It's a spcl char
```

3) If then else

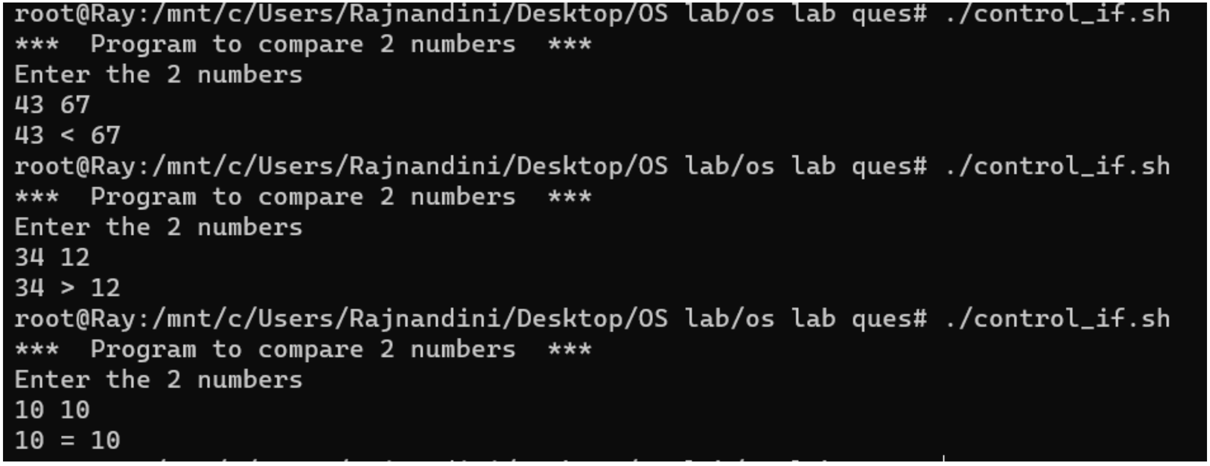
Write a shell program to compare two user input numbers.

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:



```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques# ./control_if.sh
***  Program to compare 2 numbers  ***
Enter the 2 numbers
43 67
43 < 67
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques# ./control_if.sh
***  Program to compare 2 numbers  ***
Enter the 2 numbers
34 12
34 > 12
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques# ./control_if.sh
***  Program to compare 2 numbers  ***
Enter the 2 numbers
10 10
10 = 10
```

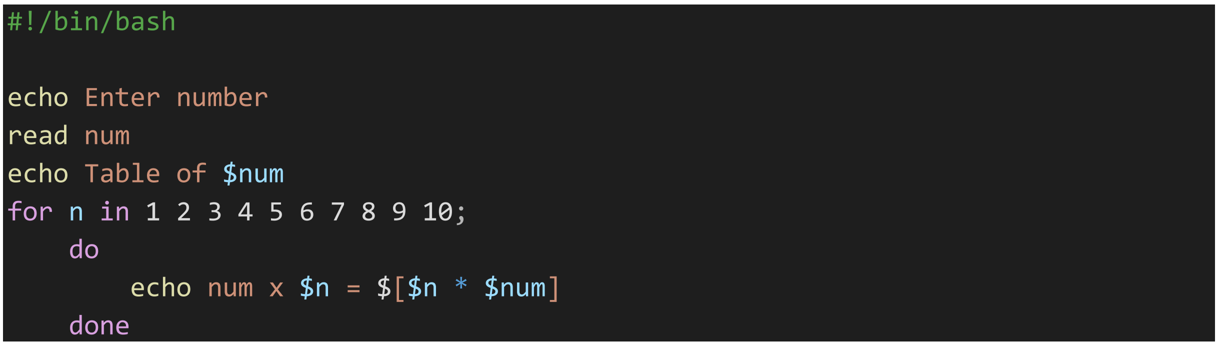
4) **LOOPS** :

A) For Loops:

1. Simple

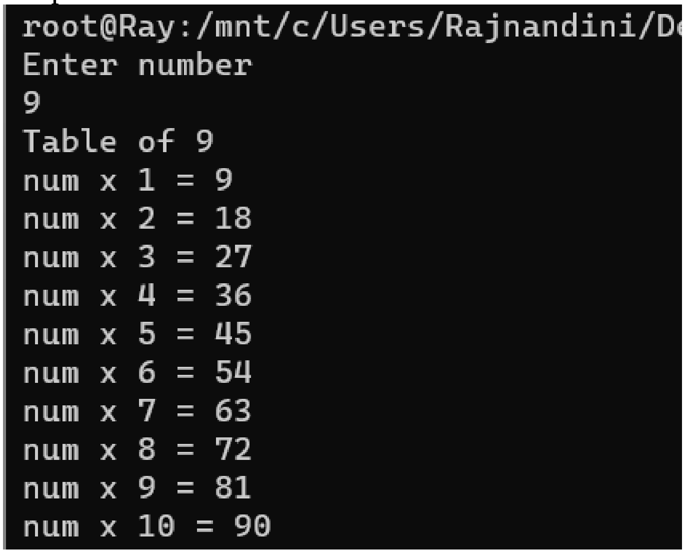
   Write a shell program to table of a number using simple for loop.

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:



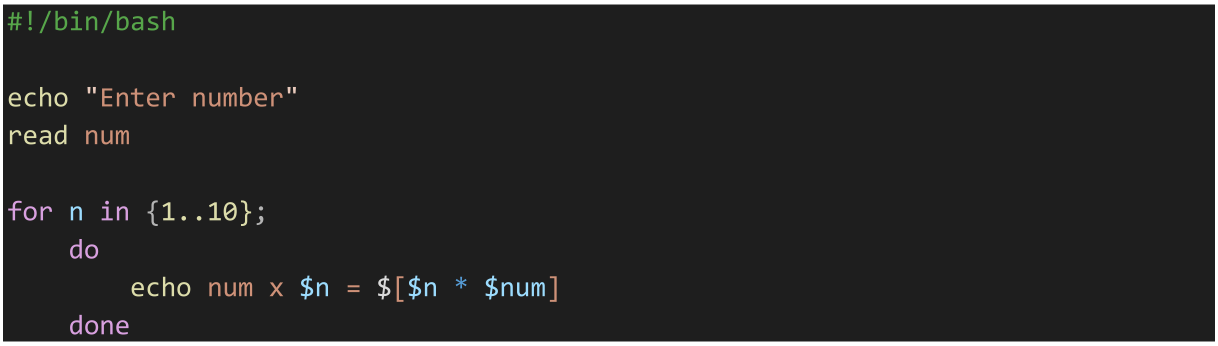
```
root@Ray:/mnt/c/Users/Rajnandini/De
Enter number
9
Table of 9
num x 1 = 9
num x 2 = 18
num x 3 = 27
num x 4 = 36
num x 5 = 45
num x 6 = 54
num x 7 = 63
num x 8 = 72
num x 9 = 81
num x 10 = 90
```

---

2. Range-based

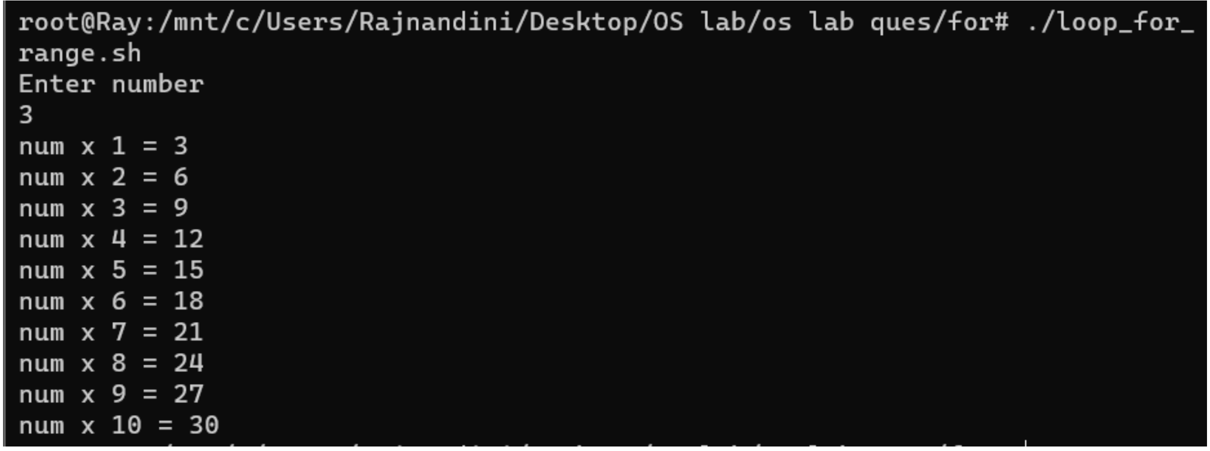
   Write a shell program to display table of user input number using range-based for loop.

   Code:



```
#!/bin/bash

echo "Enter number"
read num

for n in {1..10};
    do
        echo num x $n = $[$n * $num]
    done
```

Output:

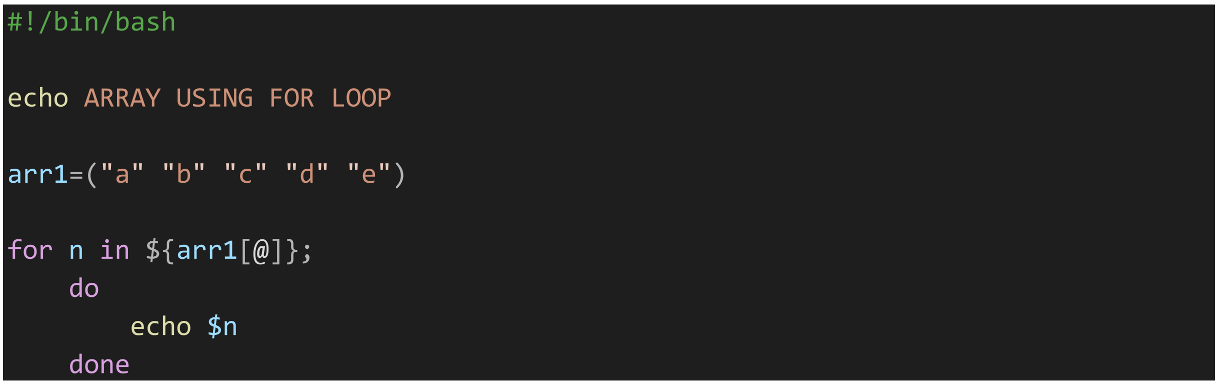


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/for# ./loop_for_
range.sh
Enter number
3
num x 1 = 3
num x 2 = 6
num x 3 = 9
num x 4 = 12
num x 5 = 15
num x 6 = 18
num x 7 = 21
num x 8 = 24
num x 9 = 27
num x 10 = 30
```

3. Array iteration

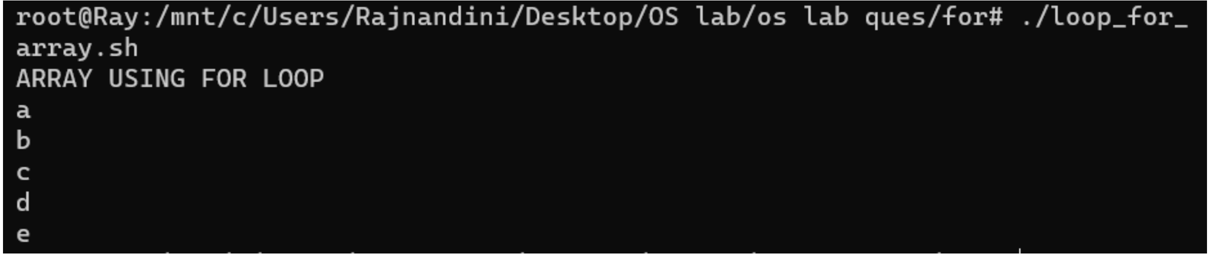
Write a shell program to implement iteration through a predefined array.

Code:



```
#!/bin/bash

echo ARRAY USING FOR LOOP

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

for n in ${arr1[@]};
    do
        echo $n
    done
```

Output:

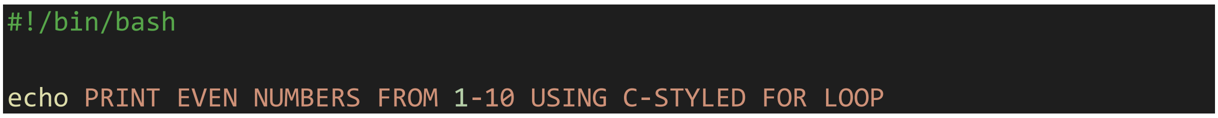


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/for# ./loop_for_
array.sh
ARRAY USING FOR LOOP
a
b
c
d
e
```

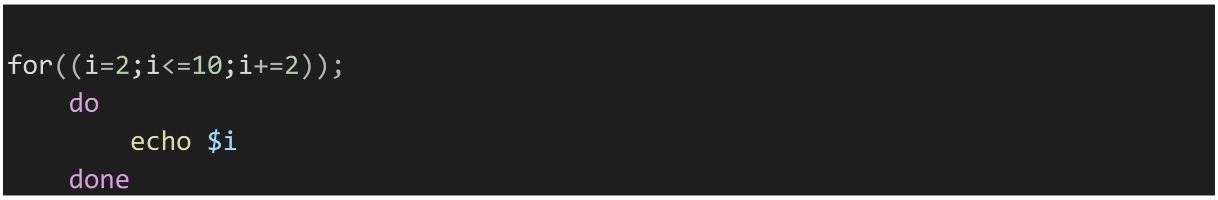
4. C-style

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

Code:

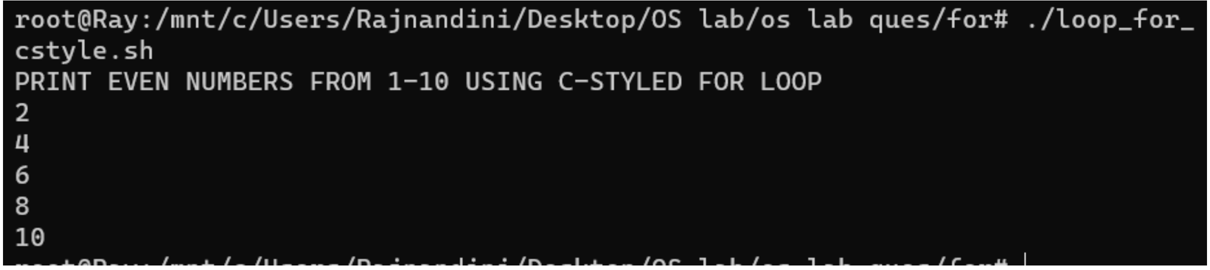


```
#!/bin/bash

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



```
for((i=2;i<=10;i+=2));
    do
        echo $i
    done
```

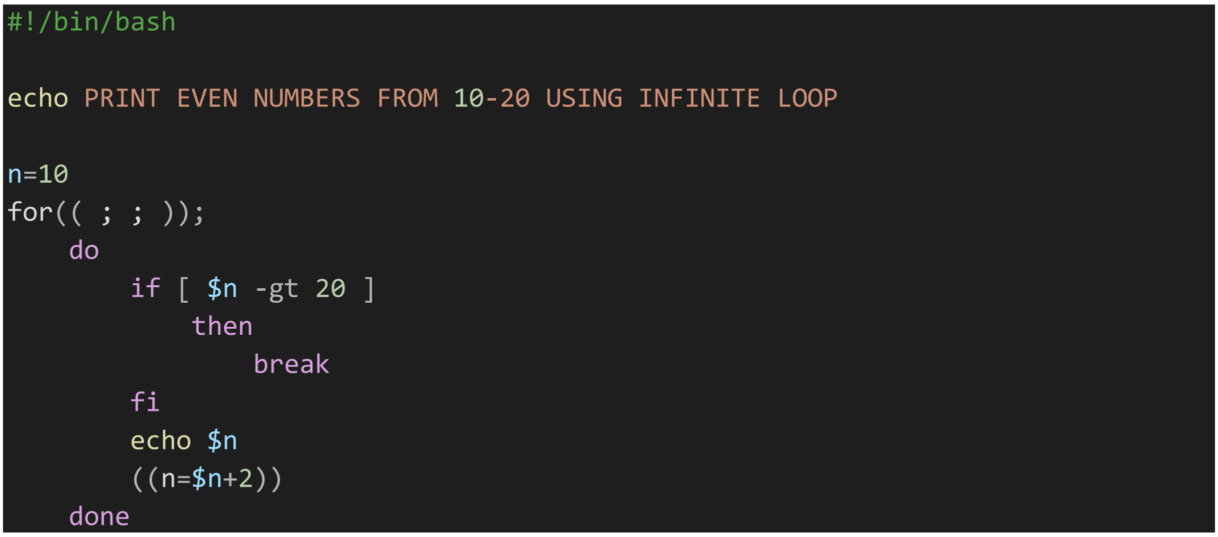
Output:



```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/for# ./loop_for_
cstyle.sh
PRINT EVEN NUMBERS FROM 1-10 USING C-STYLED FOR LOOP
2
4
6
8
10
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/for#
```

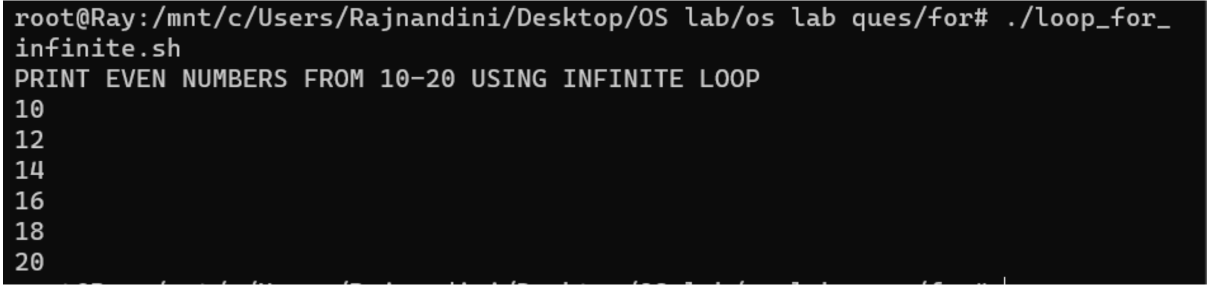
---

5. <u>Infinite</u>
   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:

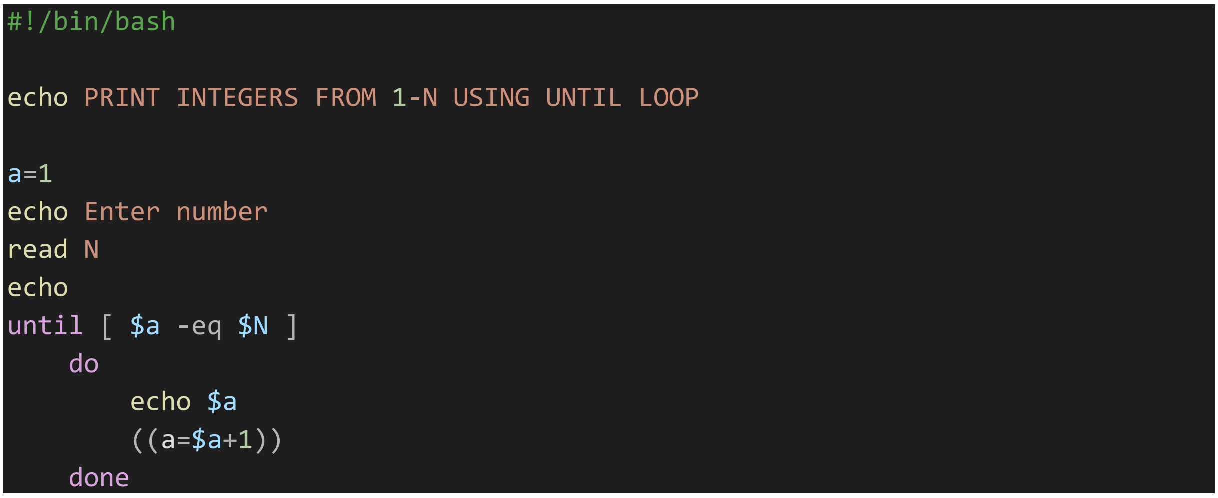


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/for# ./loop_for_
infinite.sh
PRINT EVEN NUMBERS FROM 10-20 USING INFINITE LOOP
10
12
14
16
18
20
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/for#
```

B) <u>Until loop</u>

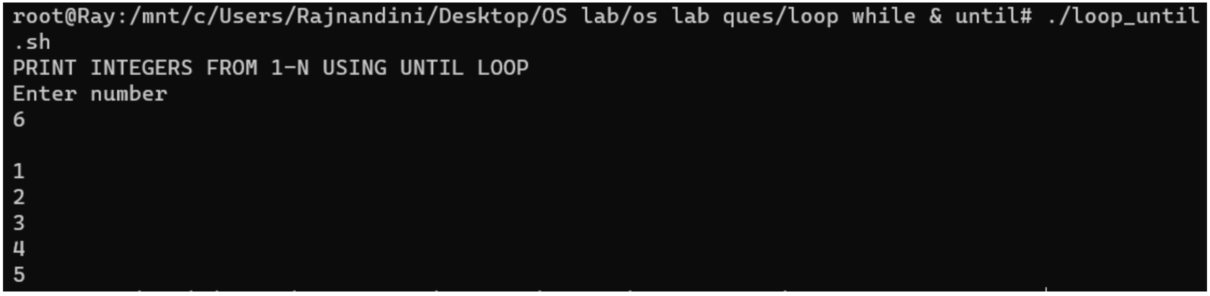
Write a shell program to print integers from 1-N using until loop taking N as input from user.

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:

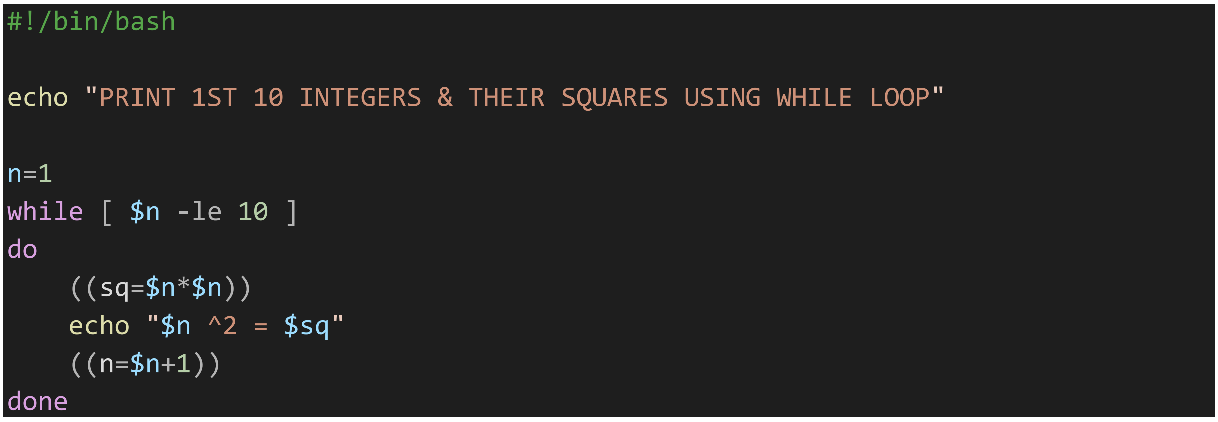


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/loop while & until# ./loop_until
.sh
PRINT INTEGERS FROM 1-N USING UNTIL LOOP
Enter number
6

1
2
3
4
5
```

C) While loop

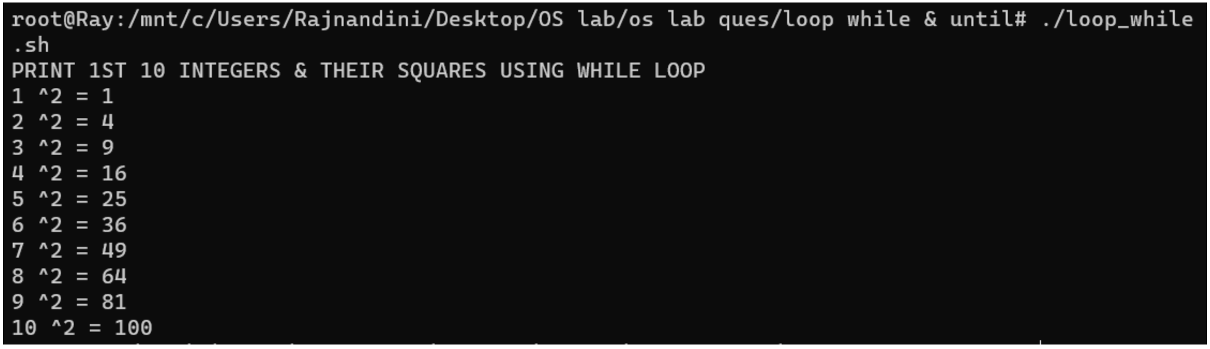
Write a shell program to print first 10 integers and their squares using while loop.

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:

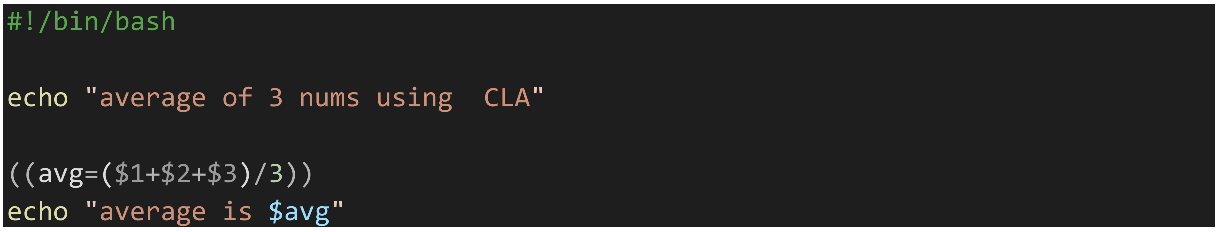


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/loop while & until# ./loop_while
.sh
PRINT 1ST 10 INTEGERS & THEIR SQUARES USING WHILE LOOP
1 ^2 = 1
2 ^2 = 4
3 ^2 = 9
4 ^2 = 16
5 ^2 = 25
6 ^2 = 36
7 ^2 = 49
8 ^2 = 64
9 ^2 = 81
10 ^2 = 100
```

5) <u>CLA</u>

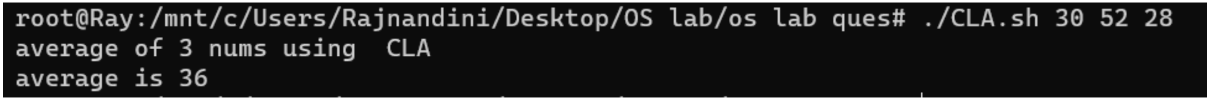
Write a shell program to find average of three numbers using command line argument.

Code:



```
#!/bin/bash

echo "average of 3 nums using  CLA"

((avg=($1+$2+$3)/3))
echo "average is $avg"
```

Output:

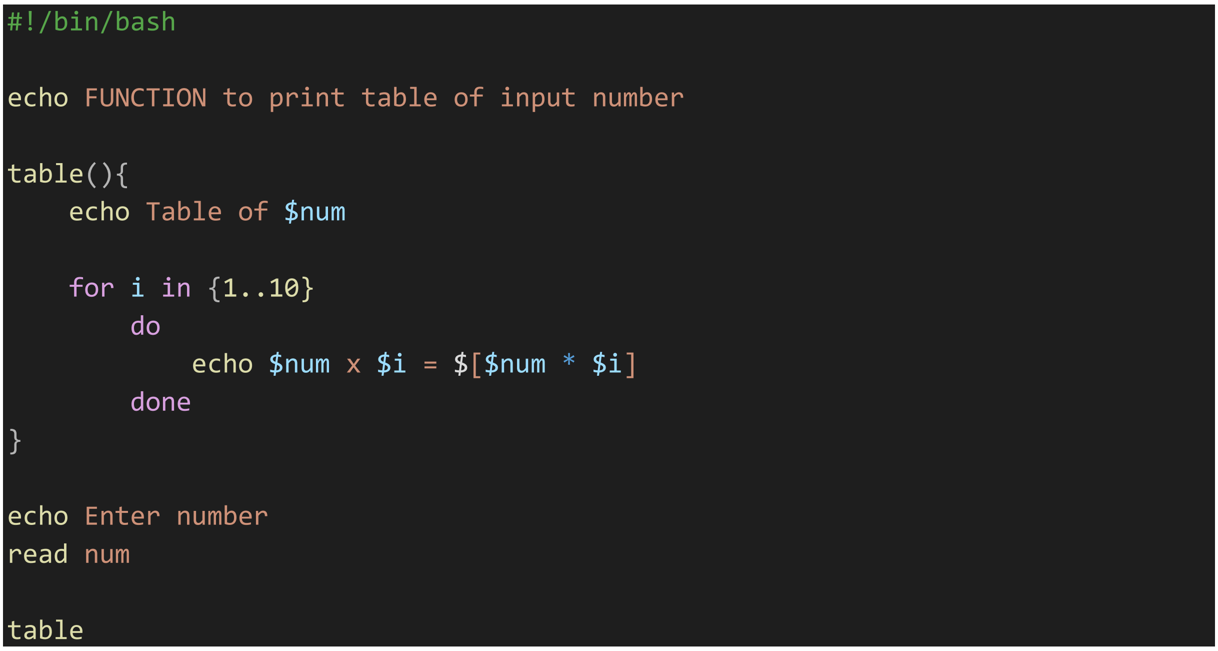


```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques# ./CLA.sh 30 52 28
average of 3 nums using  CLA
average is 36
```

6) <u>Function</u>

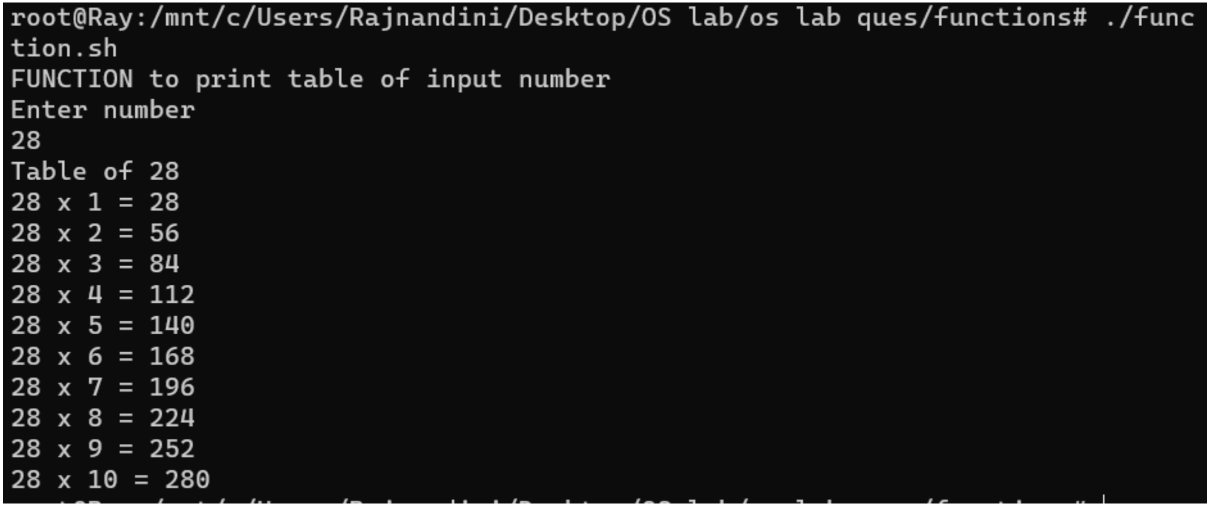
Write a shell program to print table of input number using function.

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:



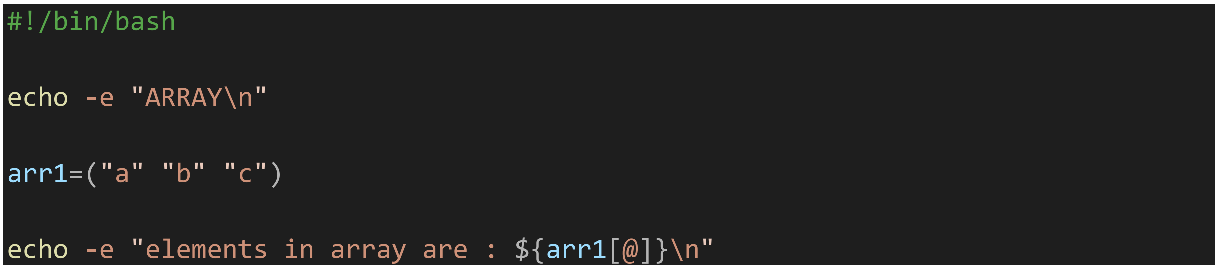
```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/functions# ./func
tion.sh
FUNCTION to print table of input number
Enter number
28
Table of 28
28 x 1 = 28
28 x 2 = 56
28 x 3 = 84
28 x 4 = 112
28 x 5 = 140
28 x 6 = 168
28 x 7 = 196
28 x 8 = 224
28 x 9 = 252
28 x 10 = 280
```

7) Array

1. Simple Array

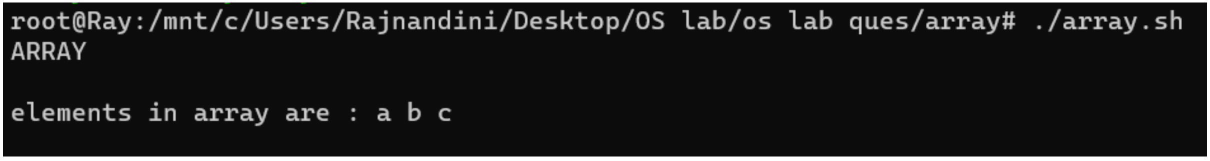
Write a shell program to display elements in an array

Code:



```
#!/bin/bash

echo -e "ARRAY\n"

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

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

Output:



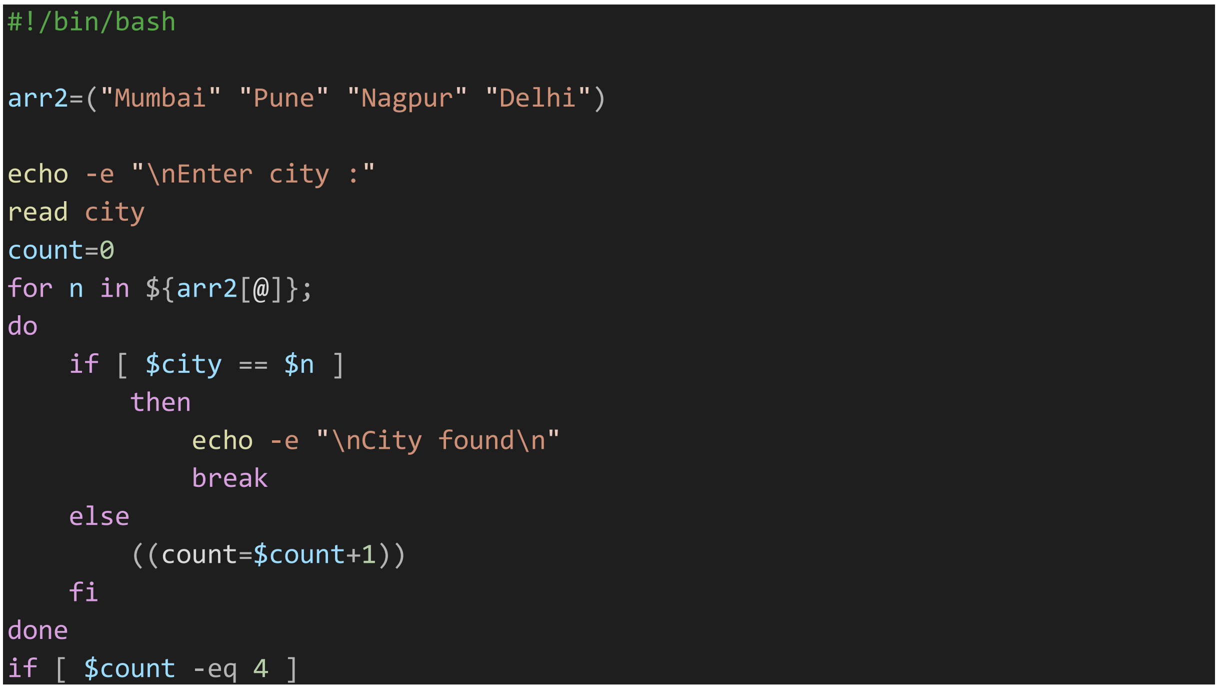
```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/array# ./array.sh
ARRAY

elements in array are : a b c
```

---

2. 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.

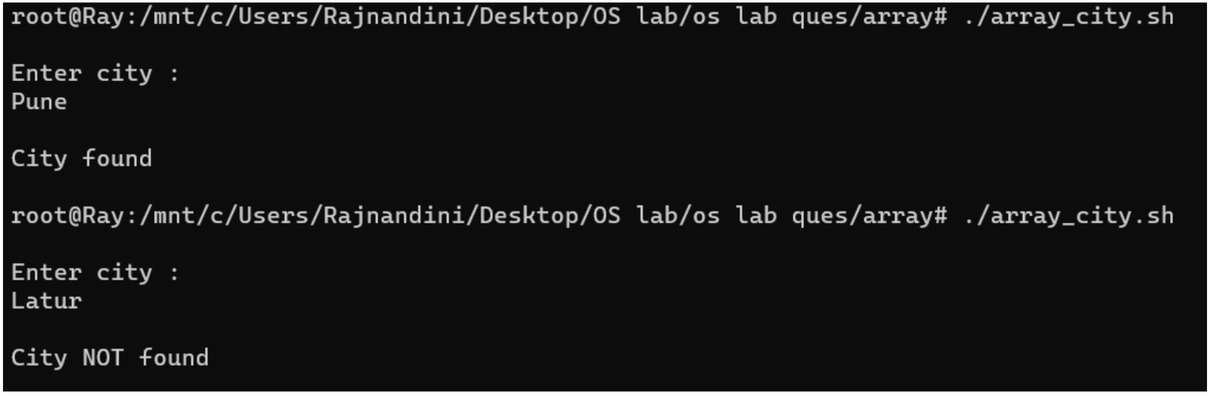
Code:



```
#!/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:



```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/array# ./array_city.sh

Enter city :
Pune

City found

root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/array# ./array_city.sh

Enter city :
Latur

City NOT found
```

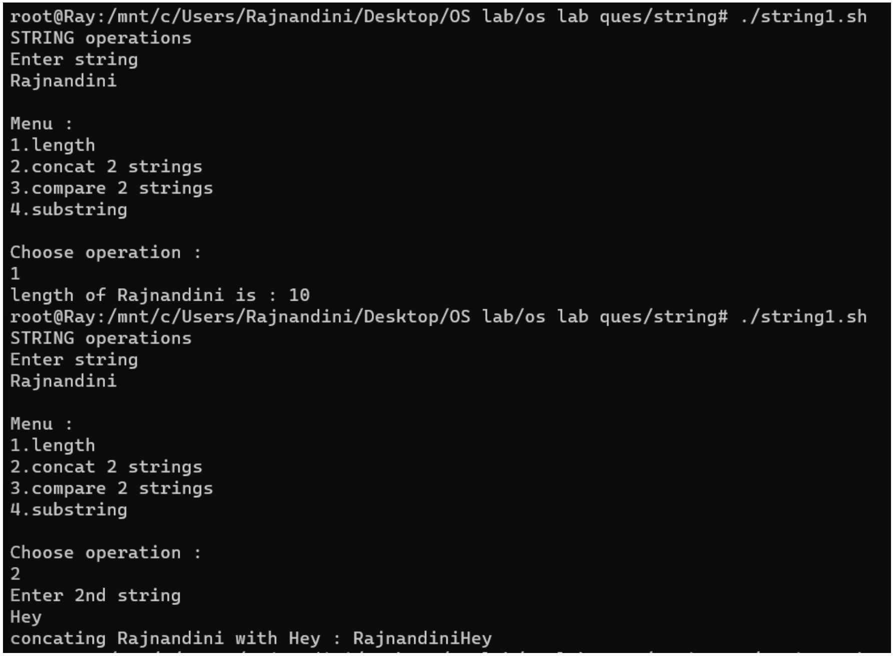
8) String (length concatenation substring compare)

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.

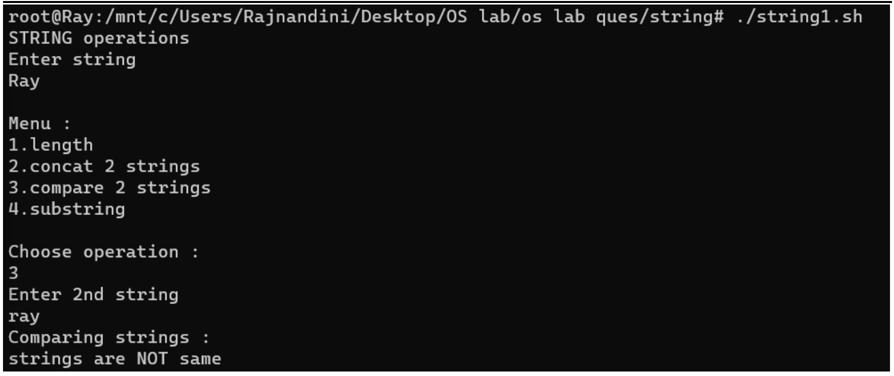
Code:

```
#!/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
```

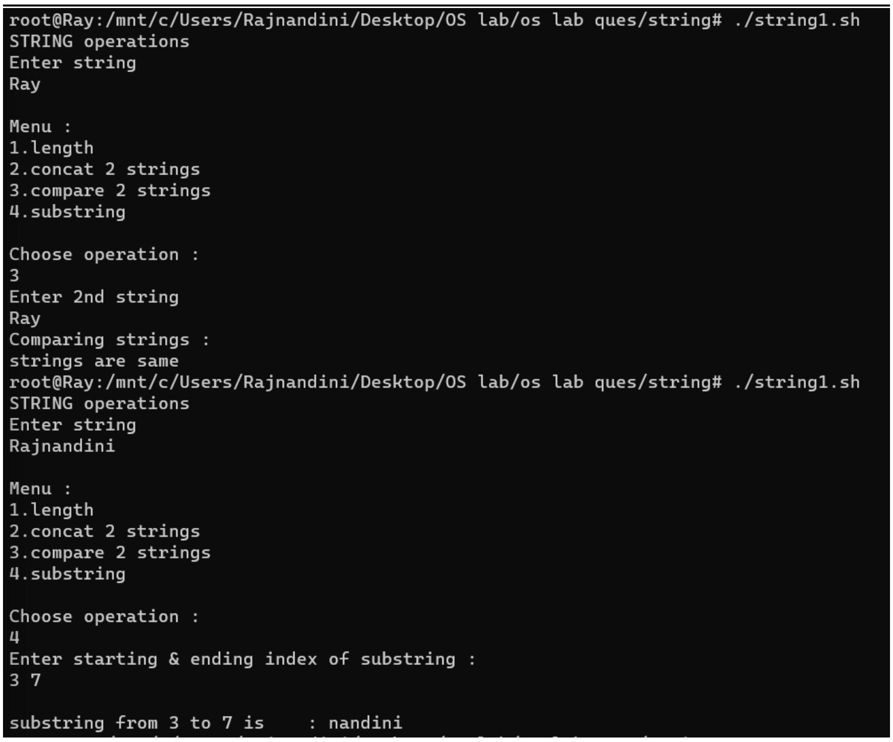
Output:



```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/string# ./string1.sh
STRING operations
Enter string
Rajnandini

Menu :
1.length
2.concat 2 strings
3.compare 2 strings
4.substring

Choose operation :
1
length of Rajnandini is : 10
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/string# ./string1.sh
STRING operations
Enter string
Rajnandini

Menu :
1.length
2.concat 2 strings
3.compare 2 strings
4.substring

Choose operation :
2
Enter 2nd string
Hey
concating Rajnandini with Hey : RajnandiniHey
```



```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/string# ./string1.sh
STRING operations
Enter string
Ray

Menu :
1.length
2.concat 2 strings
3.compare 2 strings
4.substring

Choose operation :
3
Enter 2nd string
ray
Comparing strings :
strings are NOT same
```



```
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/string# ./string1.sh
STRING operations
Enter string
Ray

Menu :
1.length
2.concat 2 strings
3.compare 2 strings
4.substring

Choose operation :
3
Enter 2nd string
Ray
Comparing strings :
strings are same
root@Ray:/mnt/c/Users/Rajnandini/Desktop/OS lab/os lab ques/string# ./string1.sh
STRING operations
Enter string
Rajnandini

Menu :
1.length
2.concat 2 strings
3.compare 2 strings
4.substring

Choose operation :
4
Enter starting & ending index of substring :
3 7

substring from 3 to 7 is    : nandini
```