TAPI DIPLOMA ENGINEERING COLLEGE, SURAT

COMPUTER ENGENEERING DEPARTMENT

SUBJECT: Scripting Language - Python (4330701)

VIRTUAL LAB

Python Programming Lab

INTRODUCTION

In this lab, students will be able to learn and practice basic python programming. Students can expand their skillset by learning and solving basic problems in python. Python is very easy to use, powerful, and versatile. It has become the language of choice for many IoT developers. One of the main reasons for the popularity of Python is the developer community. Python developers have created and made available many specific modules that can be imported into any program to immediately lend added functionality. Python is a high-level programming language. It is easy for beginners to learn and widely utilized in many scientific areas for data assessment. This lab is an overview of the Python programming language for learners without prior programming knowledge. It follows the concept of object-oriented programming and has graphical user interface-driven applications. The examples and difficulties used in this lab are drawn from numerous areas like text processing, simple graphics creation, and image manipulation, Data Science and visualization, Desktop GUI, hypertext mark-up language, and web development.

OBJECTIVE

To introduce python programming

To understand why Python is a useful scripting language for developers.

To learn how to identify Python object types.

To define the structure and components of a Python program.

To learn how to write loops and decision statements in Python.

To learn how to write functions and pass arguments in Python.

To learn how to design object?oriented programs with Python classes.

LIST OF EXPERIMENTS

1. Arithmetic Operations
2. Built-in Functions
3. Loops

EXPERIMENT - 1

In this experiment the Student will be able to understand the basics of Arithmetic Operations used in Python programming Language with the help of a iterative simulator.

Arithmetic Operations

An operator is a symbol that tells the compiler that either a mathematical or a logical manipulation has to be done. In this lab you will be studying about the Arithmetic Operations.

They are of the following types :

Addition Operator (+)

The addition operator is used to add two numbers. It is placed between two numbers that are to be added.

Syntax : number1 + number2

Program:

```
x = 3
y = 2
print('x + y =', x + y )
```

Output:

x + y = 5

Subtraction Operator (-)

The subtraction operator is used to subtract two numbers. It is placed between two numbers that are to be subtracted. The right placed number is subtracted from the one that is placed at left.

Syntax : number1 - number2

Program:

```
x = 3
y = 2
print('x - y =', x - y )
```

Output:

x - y = 1

Multiplication Operator (*)

The multiplication operator is used to multiply two numbers. It is also placed between the two numbers that are to be operated.

Syntax : number1 * number2

Program:

```
x = 30
y = 10
print('x * y =', x * y )
```

Output:

x * y = 300

Division Operator (/)

The division operator is used to divide two numbers. It is used between the numbers that are to be operated.

Syntax : number1 / number2

It has some different rules that have to be kept in mind before operating the numbers. Python2 operates the division operator by taking the integral value.

Example : 6 / 4

Answer : This operation will be solved in Python2 by taking the integral value i.e 1. Therefore, the answer of 6 / 4 = 1

This problem can be taken care by Type Casting. Type Casting is used to convert the output in a desired form.

To get the correct answer of the above example, we will type cast it using float data type.

Program:

```
x = 30
y = 10
print('x / y =', x / y )
```

Output:

x/y = 3.0

Example : float(6 / 4)

Answer : Now, the output will be changed into float type and the answer will be 1.5.

float(6 / 4) = 1.5

There's another way of solving such problem. By using one float type input, we can get the desired answer.

Example : 6.0 / 4

Answer : 1.5

Modulus Operation (%)

It is used to give out the remainder of a division operation. It is also placed between numbers. The right placed number divides the one on the left and the remainder is given as output.

Syntax : number1 % number2

Program:

```
x = 6
y = 4
print('x % y =', x % y )
```

Output:

x % y = 2

Exponent Operation (**)

It is used to perform exponential calculations. The right placed number acts as the power.

Syntax : number1* *number2

Program:

```
x = 2
y = 3
print('x ** y =', x ** y )
```

Output:

X * * y = 8

Floor Division Operator (//)

It is used to perform floor division. This gives the result in int format.

Syntax : number1 // number2

Program:

```
x = 2
y = 3
print('x // y =', x // y )
```

Output:

X // y = 0

45 / 9 will give 5.0 where as 45 // 9 will give 5

PRE-TEST

1. What is the output of 2**3?

a: 6

b: 12

c: 8

d: 18

2. What is answer of this expression, 10 % 3 is?

a: 3.33

b: 10

c: 3

d: 1

3. What is the output of this expression,2**1**2*2**3?

a: 16

b: 32

c: 34

d: Error

4. The equal sign = is:

a: Boolean operator

b: Comparison operator

c: Assignment operator

d: Logical operator

5. Which one of the following have the same precedence?

a: * and /

b: + and -

c: Both a and b

d: None of the mentioned

PROCEDURE

Steps of simulator :

1. Read the simulator details.

2. Enter the values you want to operate.

3. Select the desired type of operation from the operations list.

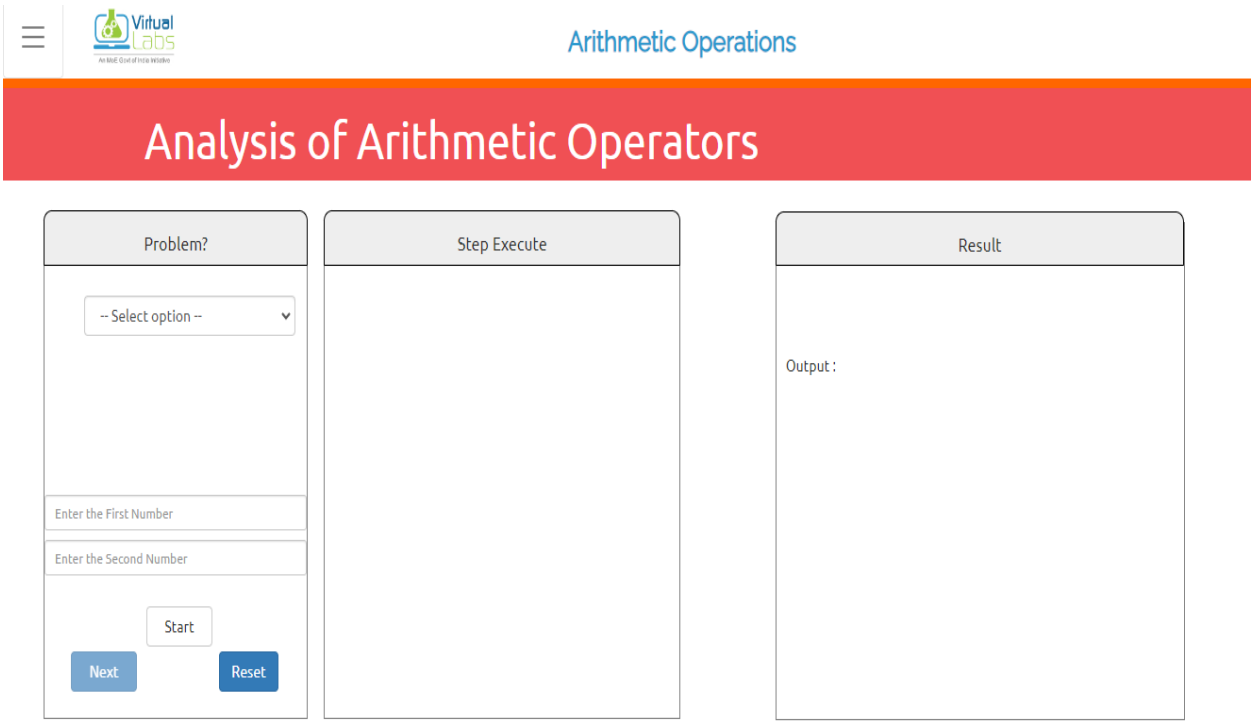
4. Press CALCULATE to proceed.

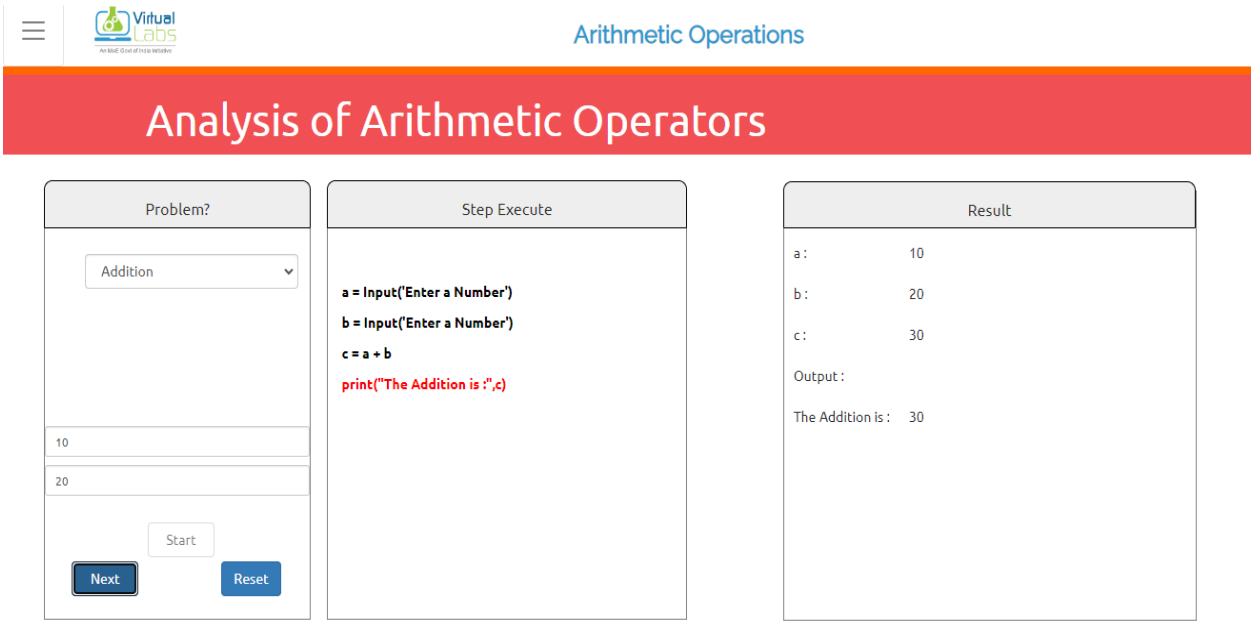
5. Press NEXT to see the execution of the code.

6. Relevant line in the code will be highlighted.

7. The local variables will be shown in the Output Panel with their values.

SIMULATION





POST-TEST

1. Is Python a case sensitive language?

a: No

b: Yes

2. What will be the output of the following code?

x= 2,7

Print(x)

a: 2

b: 7

c: (2,7)

d: Error

3. What will be the output of the following code?

x=5.5

y=5.5

Print(x+y)

a: 11

b: 11.0

c: 10

d: 0

4. What will be the output of the following code?

a=25.0

b=5.0

Print(a%b)

a: 5

b: 5.0

c: 0

d: 0.0

5. What will be the output of the following code?

```
i=47
j=25
i=i%10
j=j%10
Print(i*j)
```

a: 8

b: 53

c: 35

d: 72

EXPERIMENT - 2

In this experiment the Student will be able to understand the basics of functions used in Python programming language with the help of a iterative simulator.

THEORY

Built-in Functions

An executable program in a programming language contains multiple lines. To simplify this code, various functions are used. They can be built-in functions or user defined functions.A function is basically a chunk or module of code that takes in some input from the user and may or may not give any output. The function may provide some alterations to the input values.

'id' function :

This function returns the identity of an object. A identity has to be unique and constant for a particular object during the lifetime.

Syntax : id(object)

Program:

```
str1 = "hello"
print(id(str))
```

Output

1750812191152

Program:

```
str1 = "hello"
str2 = "world"
print(id(str1)==id(str2))
```

Output:

False

'Type' function :

This function returns the data type of an object. It returns the following data types :

i. Integer

ii. String

iii. Float

type() method returns class type of the argument(object) passed as parameter. type() function is mostly used for debugging purposes.

Syntax : type(object)

Program:

```
# Class of type dict
class DictType:
    DictNumber = {1:'apple', 2:'kiwi',
                  3:'orange', 4:'grapes'}

    # Will print the object type
    # of existing class
    print(type(DictNumber))
```

Output:

<class 'dict'>

Program:

```
str1 = 20.5
a = "hello"
b = 10
print(type(str1))
print(type(a))
print(type(b))
```

Output:

<class 'float'>

<class 'str'>

<class 'int'>

PRE-TEST

1. Which of the following functions is a built-in function in python?

a: seed()

b: sqrt()

c: print()

d: factorial()

2. What is the output of the expression?

round(2.578)

a: 2.6

b: 2.5

c: 2

d: 3

3. What is the output of the following function ?

hex(10)

a: 0xA

b: 0xa

c: 0Xa

d: a

4. What is the output of the function shown below?

```
import math
print(math.floor(5.5))
```

a: 5.5

b: 6

c: 5

d: 6.0

5. What is the output of the function shown below?

```
import math
print(abs(math.sqrt(16)))
```

a: Error

b: -4

c: 4.0

d: 4

PROCEDURE

Steps of simulator :

1. Read the simulator details.

2. Enter your choice.

3. Press CALCULATE to proceed.

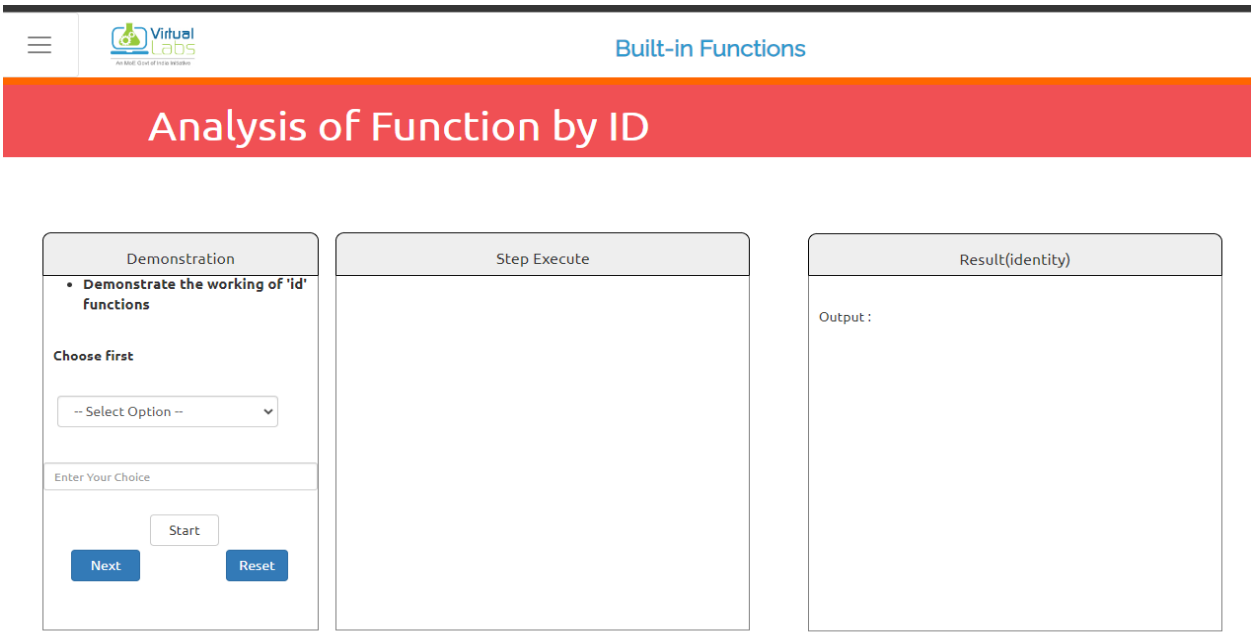
4. The code will be displayed.

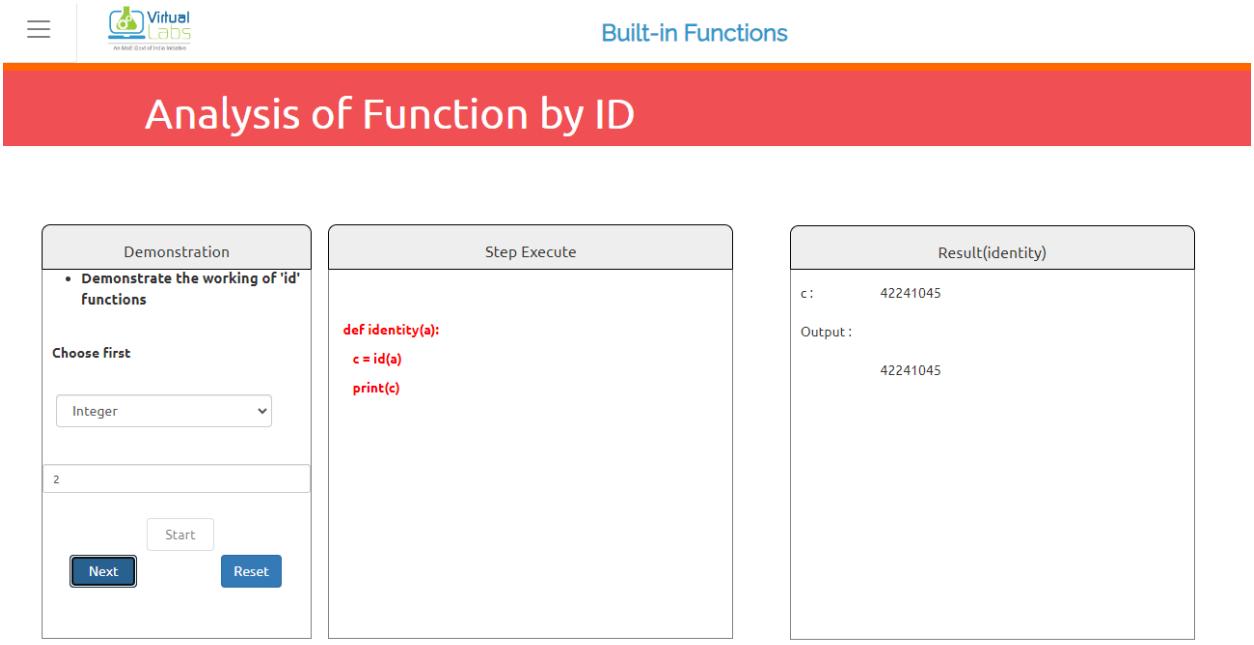
5. Press NEXT to see the execution of code.

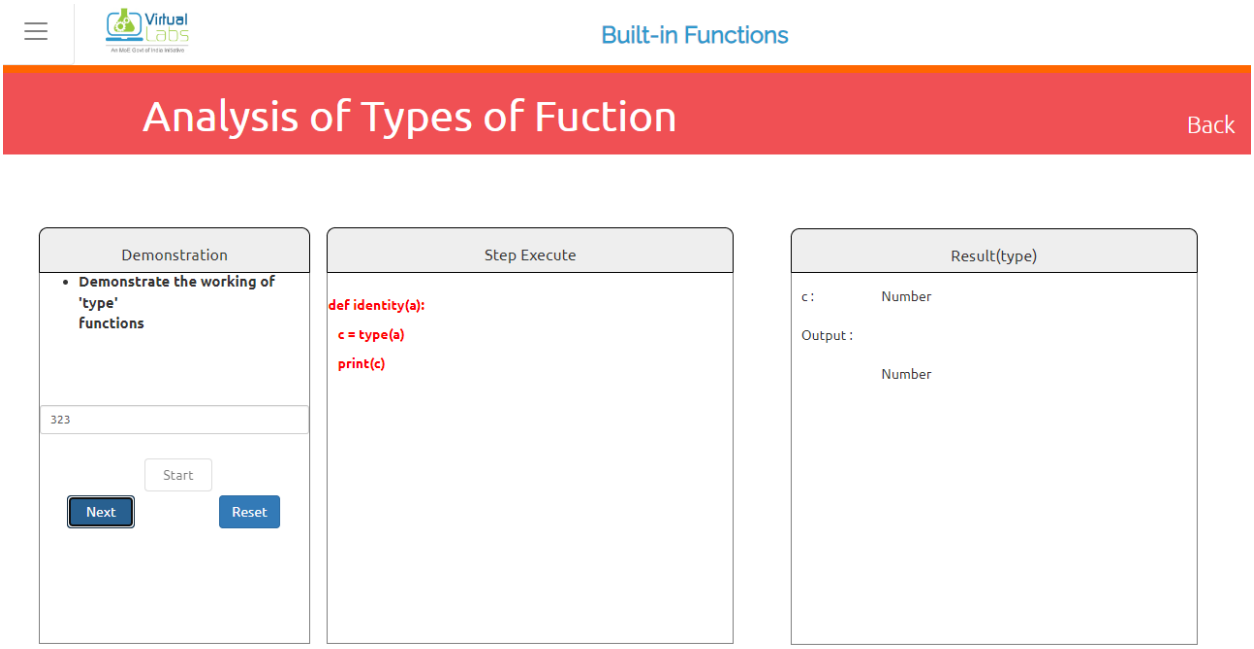
6. Relevant line in the code will be highlighted.

7. The local variables will be shown in the Output Panel with their values.

SIMULATION







POST-TEST

1. Which of the following functions will not result in an error when no arguments are passed to it?

a: any()

b: float()

c: max()

d: divmod()

2. What will be the output of the following code?

Print(len(['hii',0, 2, 4]))

a: 3

b: 6

c: 4

d: Error

3. What will be displayed by print(ord(`a') - ord(`b')) ?

a: 1

b: -1

c: 2

d: 0

4. What will be displayed by 'helloworld'.replace('l', 'e')?

a: heelowored

b: heeeoworld

c: hllloworld

d: heeeowored

5. What will be displayed by min(max(False,-2,-6), 3,5,6) ?

a: 2

b: -3

c: False

d: -4

EXPERIMENT - 3

In this experiment the Student will be able to understand the flow of controls and types of loops used in Python programming language with the help of a iterative simulator.

THEORY

LOOP

The execution of programming language codes is done by a compiler. A compiler is given a set of codes or rather a sequence of codes that perform a desired task. The task may or may not be repetitive but the compiler is smart enough to process it. Such repetitive code is known as a 'loop'.

Loop is a sequential set of instructions which gets executed multiple times to reduce minimize the repetition of code.

In Python, we have two types of loops :

i. for loop

ii. while loop

To understand the functioning and flow of a loop, you must get familiar with the term 'block'. A block is the smallest unit in a loop which performs one particular task.

'For' loop :

Syntax :

for object in range(initialization, limit, update):

statements

The above given syntax is of for loop where we put the object name after 'for' and the limit inside 'range( )'.

Program:

```
# Python program to illustrate
# Iterating over range 0 to n-1

n = 4
for i in range(0, n):
    print(i)
```

Output :

0

1

2

3

4

'While' loop :

Syntax :

while expression:

statements

The above statement is for while loop, where the testing condition is placed after while and it is followed by the statements placed in the loop body.

Program:

```
# Python program to illustrate
# while loop
count = 0
while (count < 3):
    count = count + 1
    print("Hello people")
```

Output:

Hello people

Hello people

Hello people

Program:

```
# Python program to illustrate
# Single statement while block
count = 0
while (count == 0): print("Hello Geek")
```

Output:

hello geek

hello geek

.....

Nested loops :

A nested loop is an inner loop in the loop body of the outer loop. The inner or outer loop can be any type, such as a while loop or for loop. For example, the outer for loop can contain a while loop and vice versa

Program:

```
# Python program to illustrate
# nested for loops in Python
for i in range(1, 5):
    for j in range(i):
        print(i, end=' ')
    print()
```

Output:

1

2 2

3 3 3

4 4 4 4

Program:

```
i=1
while i<=2 :
    print(i,"Outer loop is executed only once")
    j=1
    while j<=2:
        print(j,"Inner loop is executed until to completion")
        j+=1
    i+=1;
```

Output:

1 Outer loop is executed only once

1 Inner loop is executed until to completion

2 Inner loop is executed until to completion

2 Outer loop is executed only once

1 Inner loop is executed until to completion

2 Inner loop is executed until to completion

PRE-TEST

1.What data type does the range() function generate?:

a: int

b: float

c: list

d: String

2. Which do we access key or the value while looping through a dictionary?

a: key

b: key and value in tuple format

c: key and value in list format

d: value

3. Which of the following function returns a randomly selected element from range?

a: choice(seq)

b: randrange([start,stop,step])

c: random()

d: seed()

4. Which of the following keyword is a valid placeholder for body of the function ?

a: continue

b: break

c: pass

d: body

PROCEDURE

Steps of simulator :

1. Read the simulator details.

2. Enter the values you want to proceed with.

3. Press CALCULATE to proceed.

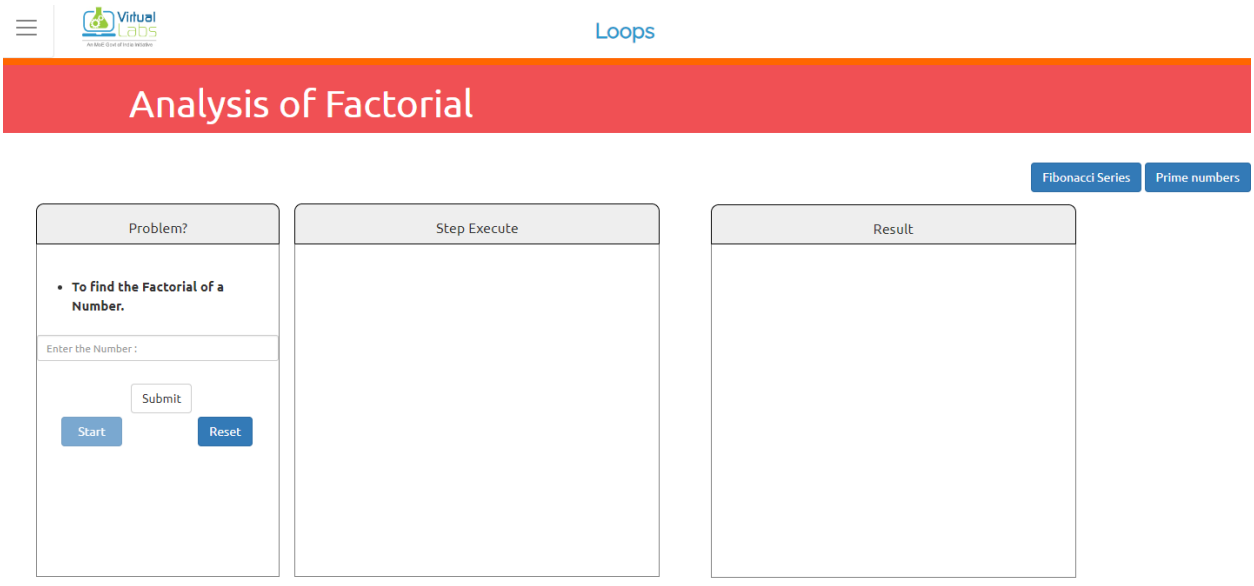
4. The code will be displayed.

5. Press NEXT to see the execution of code.

6. Relevant line in the code will be highlighted.

7. The local variables will be shown in the Output Panel with their values.

SIMULATION



Loops

Analysis of Factorial

Fibonacci Series | Prime numbers

Problem?

- **To find the Factorial of a Number.**

5

Submit

Start | Reset

Step Execute

```
for num in range(start, end + 1):
  factorial = 1
    if num < 0:
      print("Factorial for negative numbers does not exist!!!")
    elif num == 0:
      print("The factorial of the Number is :")
    else:
      for i in range(1,num + 1):
        factorial = factorial*i
      print(The Factorial of the Number is :,num)
```

Result

The Factorial of the Number is :

120

Loops

Analysis of Fibonacci Series

Back

Problem?

- **To find the Fibonacci Series.**

5

Submit

Start | Reset

Step Execute

```
n1 = 0
n2 = 1
count = 0
if nterms <= 0:
  print("Please enter a positive integer")
elif nterms == 1:
  print("Fibonacci sequence upto",nterms,":")
  print(n1)
else:
  print("Fibonacci sequence upto",nterms,":")
  while count < nterms:
```

Result

Fibonacci sequence upto nterms is :

0,1,1,2,3

POST-TEST

1.Which of the following functions will not result in an error when no arguments are passed to it?

a: any()

b: float()

c: max()

d: divmod()

2.What will be the output of the following code? for i in ''.join(reversed(list('virtual'))): print(i)

a: v i r t u a l

b: Error

c: l a u t r i v

d: None of these

3.What is the output of the following? for i in [a, b, c, d][::-1]: print (i)

a: a b c d

b: Error

c: None

d: d c b a