K. J. Somaiya College of Engineering, Mumbai-77

Batch: C2-2	**Roll No.: 047**
Experiment / assignment / tutorial No.	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Write a program to demonstrate the use of User-defined functions in Python

AIM: 1) Write a Python program using a recursive function that takes a string as input from the user and displays whether the string is Palindrome or not.

2) Write a Python program for a character frequency counter function that takes a list of strings from the user as input and displays the frequency of each character in the list.

.

OUTCOME: Students will be able to

CO1: Formulate problem statement and develop the logic (algorithm/flowchart) for its solution.

CO3: Use different Decision Making statements and Functions in Python.

Use of input output function, Use different Decision Making statements and user defined functions in Python.

Resource Needed: Python IDE

Books/ Journals/ Websites referred:

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018,India
3. https://www.geeksforgeeks.org/python-strings/?ref=lbp

Theory:

1. Python Functions

A function is a block of code that only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

Creating a Function:

In Python, a function is defined using the def keyword:

Example:
```
def my_function():
    print("Hello from a function")
```

Arguments:

Information can be passed into functions as arguments. Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

Parameters or Arguments:

The terms parameter and argument can be used for the same thing: information that is passed into a function. From a function's perspective: A parameter is the variable listed inside the parentheses in the function definition. An argument is the value that is sent to the function when it is called.

Number of Arguments:

By default, a function must be called with the correct number of arguments, i.e. if your function expects 2 arguments; you have to call the function with 2 arguments, not more, and not less.

Keyword Arguments

You can also send arguments using the key== value syntax.
This way, the order of the arguments does not matter.

**Arbitrary Keyword Arguments, **

If you do not know how many keyword arguments will be passed into your function, add two asterisks (**) before the parameter name in the function definition.
This way the function will receive a dictionary of arguments, and can access the items accordingly.

Default Parameter Value

The following example shows how to use a default parameter value.
If we call the function without argument, it uses the default value:

Passing a List as an Argument

You can send any data type of argument to a function (string, number, list, dictionary, etc.), and it will be treated as the same data type inside the function.

Return Values

To let a function return a value, use the return statement:

The pass statement

Function definitions cannot be empty, but if you, for some reason, have a function definition with no content, put it in the pass statement to avoid getting an error.

2. Recursion Function

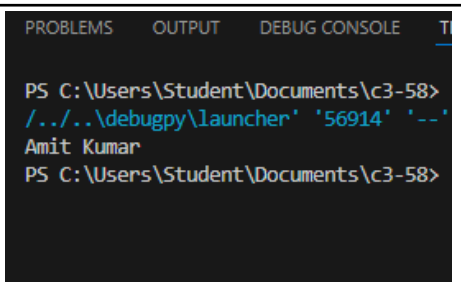
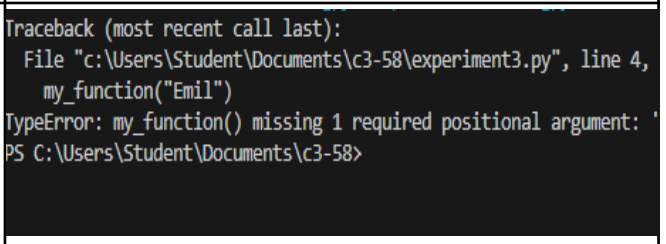
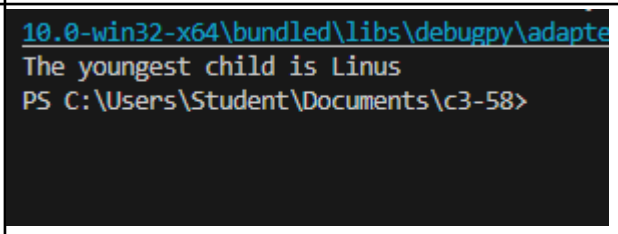
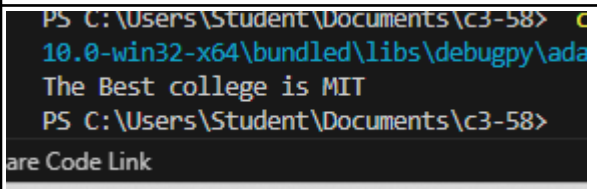
Python also accepts function recursion, which means a defined function can call itself. Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

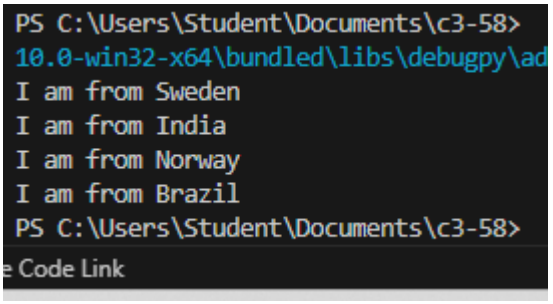
The developer should be very careful with recursion, as it can be quite easy to slip into writing a function that never terminates, or one that uses excess amounts of memory or processor power. However, when written correctly, recursion can be a very efficient and mathematically elegant approach to programming.

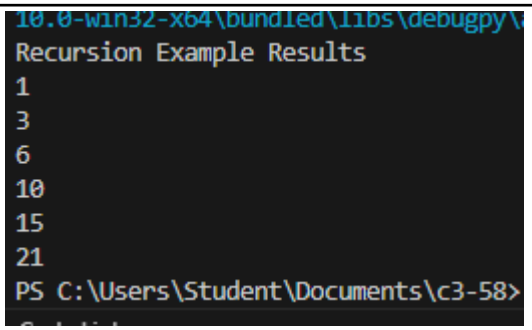
To a new programmer, it can take some time to work out how exactly this works, best way to find out is by testing and modifying it.

Problem Definition:

In the below table, the input variable, Python code, and output column is given. You have to complete a blank cell in every row.

Python Code	Output
def my_function(fname,lname): print(fname+ " " + lname) my_function("Amit", "Kumar")	
def my_function(fname, lname): print(fname + " " + lname) my_function("Emil")	
def my_function(*kids): print("The youngest child is " + kids[2]) my_function("Emil", "Tobias", "Linus")	
def my_function(college3, college2, college1): print("The Best college is " + college3) my_function(**"MIT", "Stanford", "Harvard"**)	

def my_function(**country= "Norway"**): print("I am from " + country) my_function("Sweden") my_function("India") my_function() my_function("Brazil")	

def tri_recursion(k):
if(k > 0):
result = k + tri_recursion(k - 1)
print(result)
else:
result = 0
return result
print("Recursion Example Results")
tri_recursion(6)
	

3. Write a Python program using a recursive function that takes a string as input from the user and displays whether the string is Palindrome.
4. Write a Python program for a character frequency counter function that takes a list of strings from the user as input and displays the frequency of each character in the list.

Implementation details:

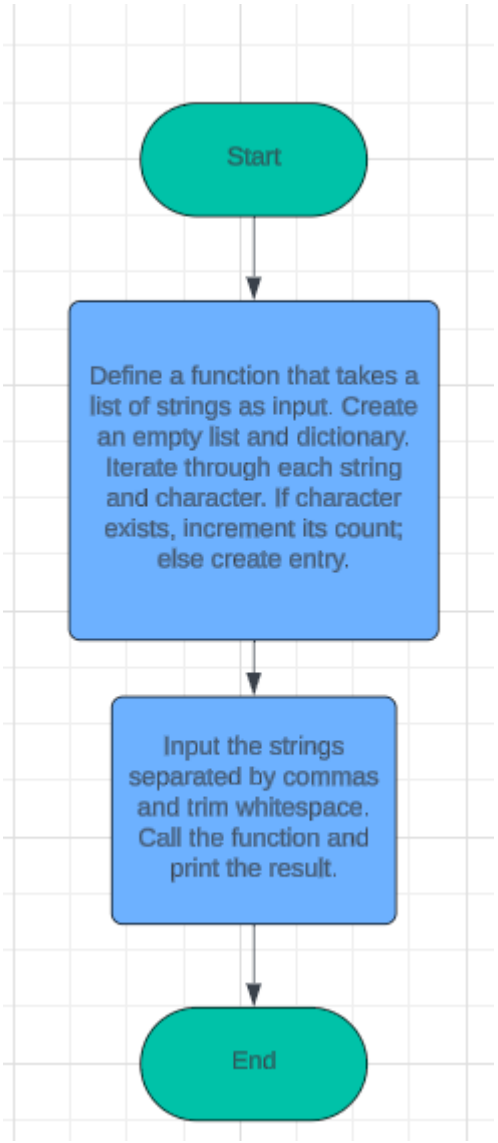
Problem 3:
Algorithm:

1. Start
2. Define a function which takes a string as input
3. If the length of the string is less than one, it is automatically palindrome. Return True.
4. If the first index of the string is not equal to the last, the string is not palindrome. Return false.
5. Call the function again by removing the first and last indices to keep comparing till the middlemost indices are checked.
6. Take an input, if the function returns true, print that the string is Palindrome.

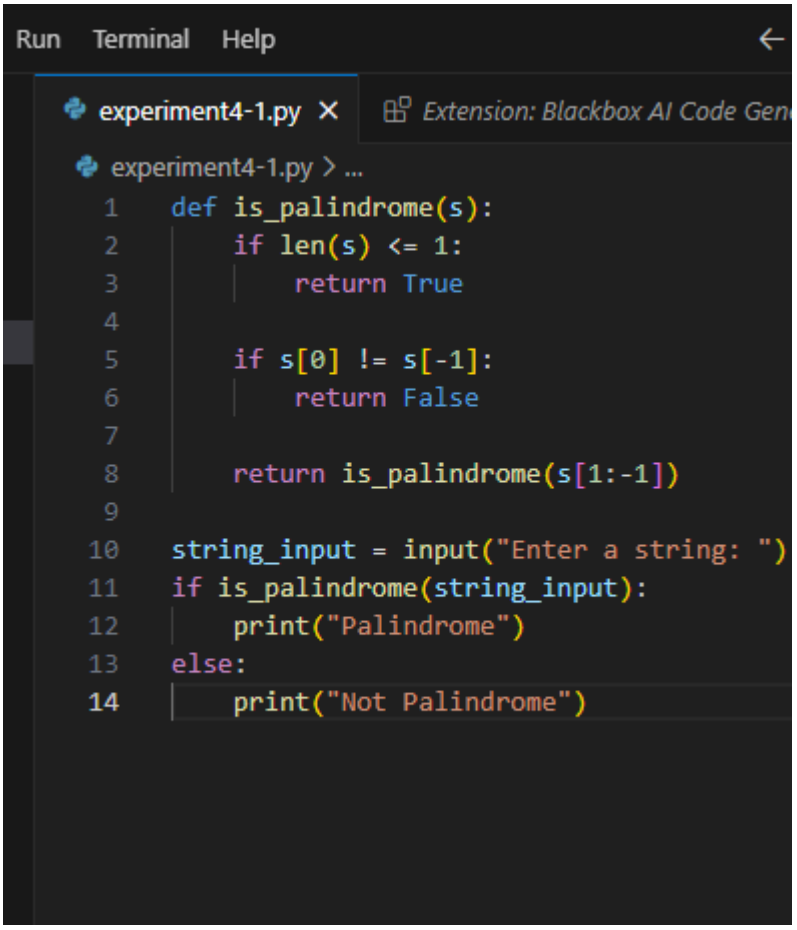
Else, print that the string is not Palindrome.

7. Stop.

Flowchart:



Code:

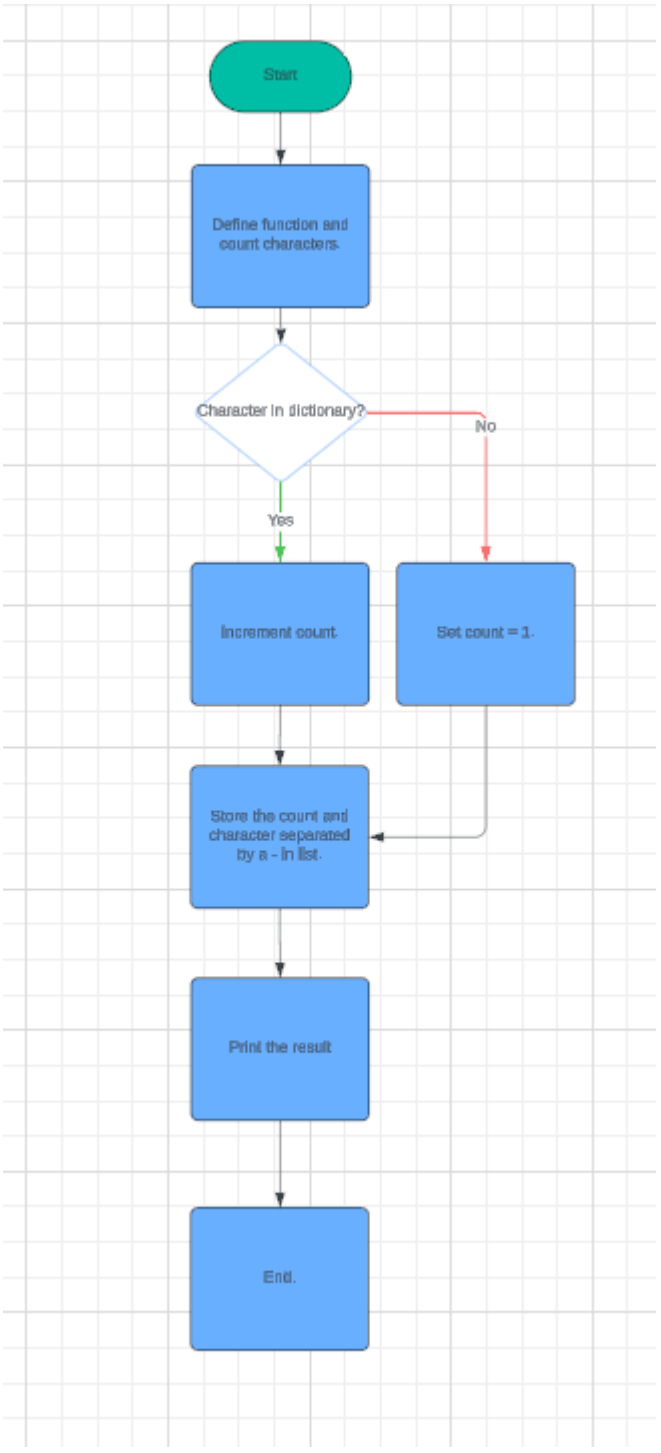


Problem 4:

Algorithm:

1. Start
2. Define a function that takes a list of strings as input.
3. Create an empty list and dictionary
4. Iterate through each string in the list of strings and through each character in each string.
5. If the empty dictionary already has that character, increment the dictionary's value. If not, then create a new dictionary item as that character and set its count to 1.
6. Run a loop that runs for each value in frequency dictionary. For each item in dictionary, store the character and its count in the list we created earlier.
7. Input the string and separate them with , using split(',’), also trim it to erase whitespaces.
8. Call the character_frequency function and pass the value of input_strings to it.
9. Print the function, which will eventually return l1, which is the list that stores characters and character counts.
10. End

Flowchart:


Start
Define function and count characters
Character in dictionary?
No
Yes
Increment count
Set count = 1.
Store the count and character separated by a - in list.
Print the result
End.

Code:

```python
def character_frequency(strings):
    l1=[]
    frequency = {}
    for string in strings:
        for char in string:
            if char in frequency:
                frequency[char] += 1
                # If we encounter same char again, we add to its counter value
            else:
                frequency[char] = 1
                # If we encounter a char for the first time, we set its count to 1

    for i in frequency:
        m=str(i)+'-'+str(frequency[i])
        l1.append(m)
    return l1

input_strings = input("Enter strings and separate them with ,: ").split(',')

input_strings = [s.strip() for s in input_strings]
# Erasing  leading and trailing whitespaces from each string

# Calculate frequency
frequency_count = character_frequency(input_strings)
print(frequency_count)
```

Output(s):

Problem 3:

```
Enter a string: hannah
Palindrome
PS C:\Users\Student\Documents\c3-58> ^C
PS C:\Users\Student\Documents\c3-58>
PS C:\Users\Student\Documents\c3-58>  c:;
10.0-win32-x64\bundled\libs\debugpy\adapt
Enter a string: Montana
Not Palindrome
PS C:\Users\Student\Documents\c3-58>
```

Problem 4:

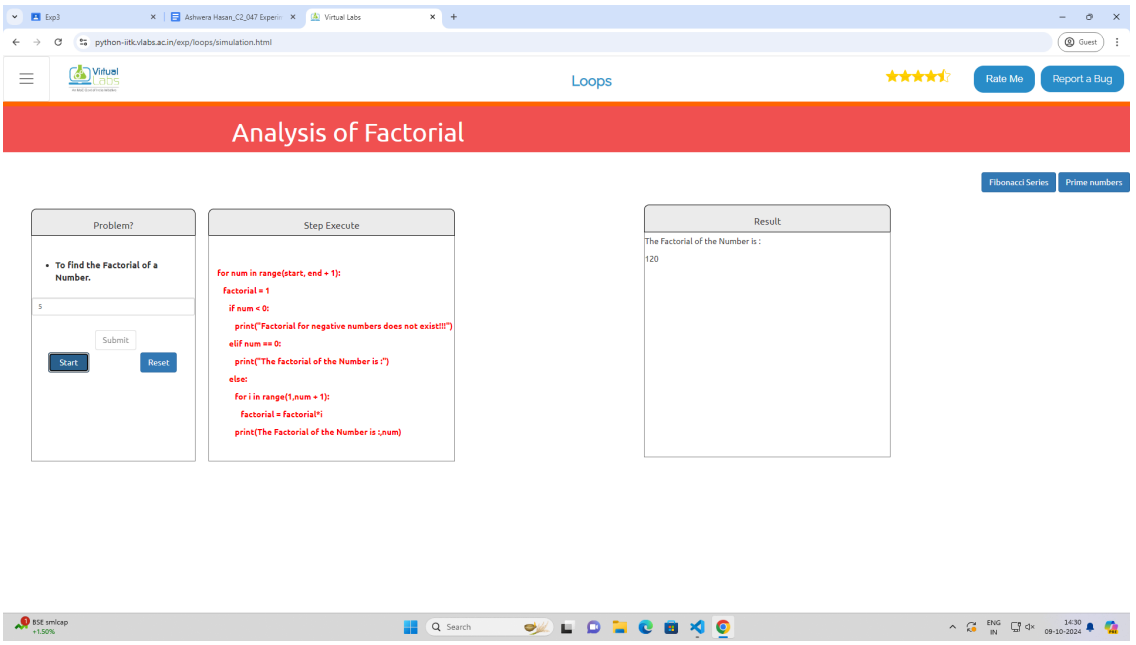
```
PS C:\Users\Student\Documents\c3-58>
PS C:\Users\Student\Documents\c3-58>  c:; cd 'c:\Users\Student\Documents\
10.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '5784
Enter strings and separate them with ,: ashwera, richa, hasan, handa
['a-7', 's-2', 'h-4', 'w-1', 'e-1', 'r-2', 'i-1', 'c-1', 'n-2', 'd-1']
PS C:\Users\Student\Documents\c3-58>
are Code Link
```

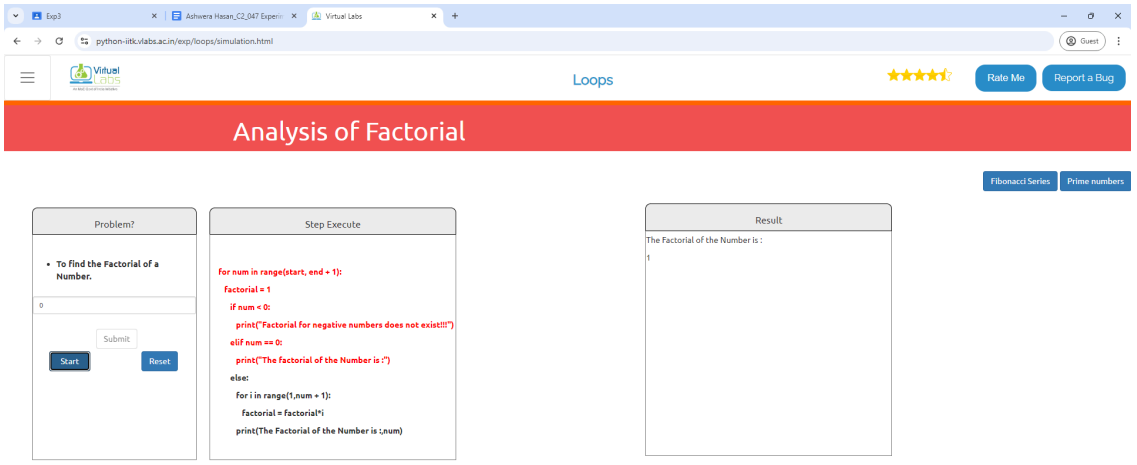
Conclusion:
Recursion in python is a powerful tool that helps with iterations, like in palindrome strings. Recursion can be used anywhere where a loop can be used. A function has a keyword “def”, followed by the function’s name. A function may or may not have a parameter and may or may not return a value.

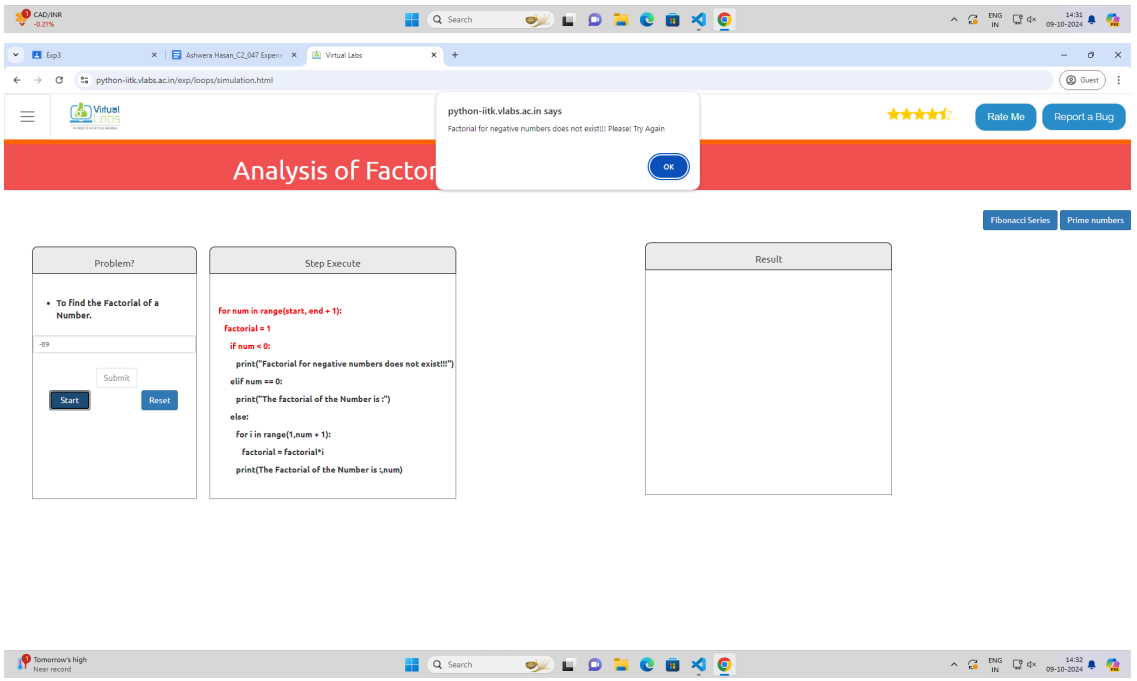
Post Lab Descriptive Questions

1. Virtual lab on Loop: https://python-iitk.vlabs.ac.in/exp/loops/
2. Virtual lab on String: https://python-iitk.vlabs.ac.in/exp/strings/
3. Write a Python program to calculate factorial using recursion
4. Define a function named 'test_range' that checks if a number 'n' is within the range 3 to 8 (inclusive)

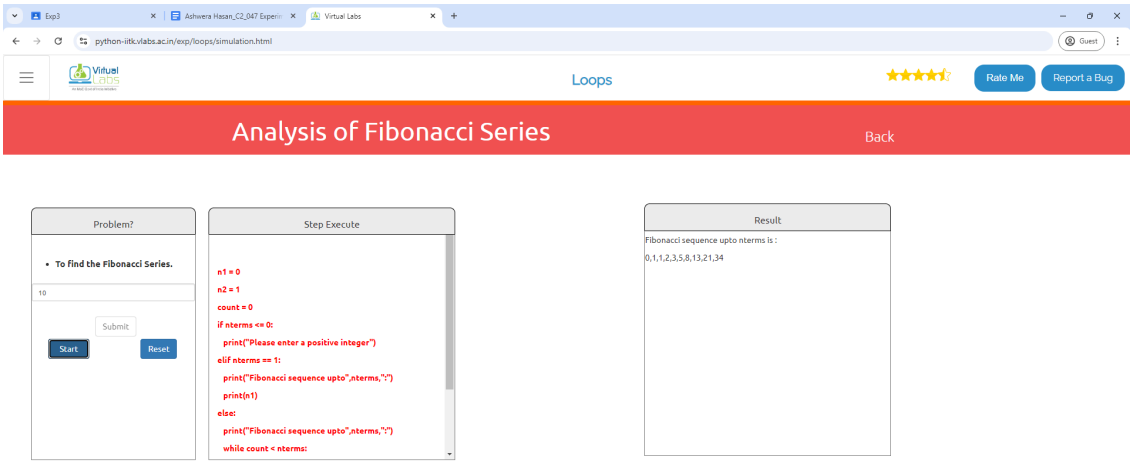
1.

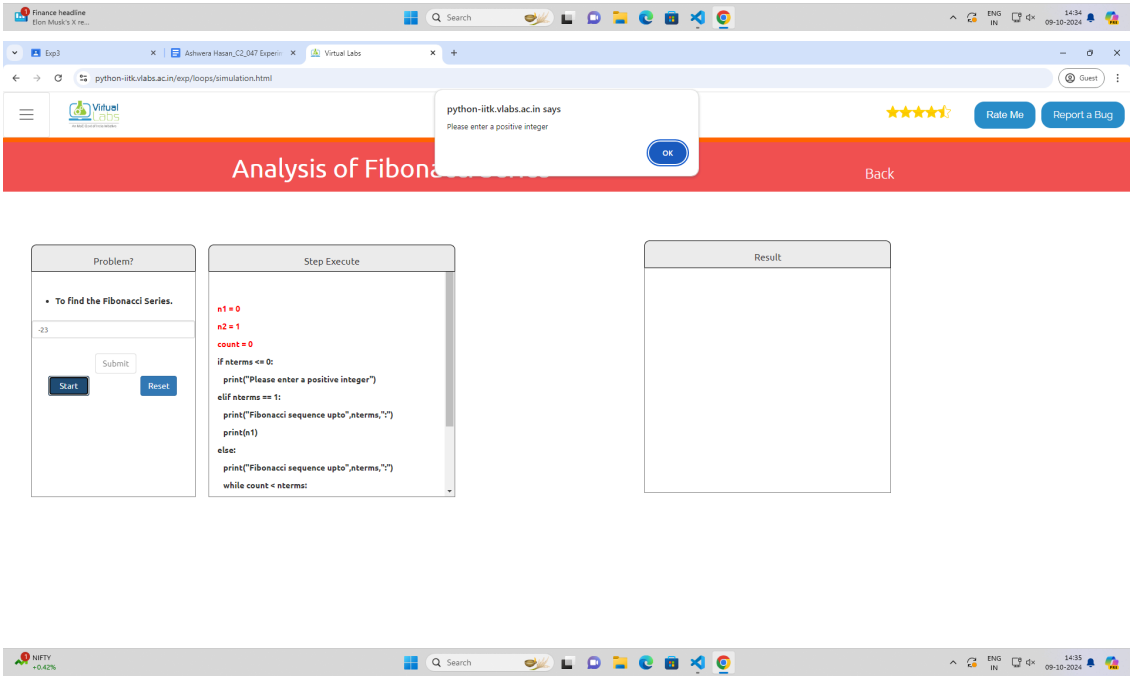


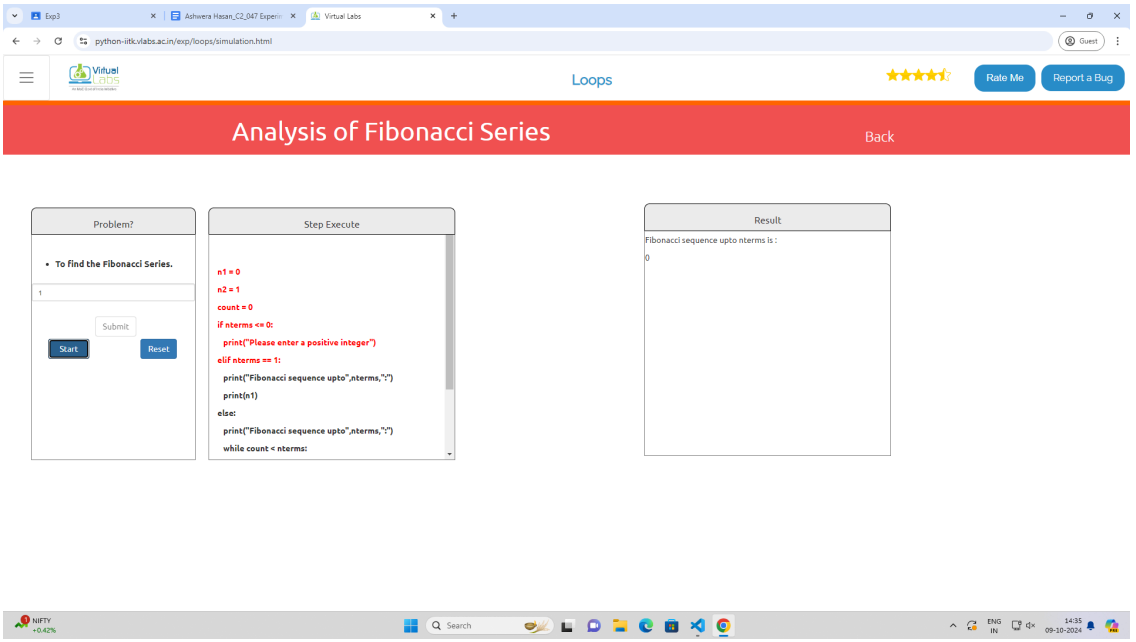

python-iitk.vlabs.ac.in/exp/loops/simulation.html
Loops
Rate Me
Report a Bug
Analysis of Factorial
Fibonacci Series
Prime numbers
Problem?
To find the Factorial of a Number.
0
Submit
Start
Reset
Step Execute
for num in range(start, end + 1):
factorial = 1
if num < 0:
print("Factorial for negative numbers 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 :
1


python-iitk.vlabs.ac.in/exp/loops/simulation.html
python-iitk.vlabs.ac.in says
Factorial for negative numbers does not exist!!! Please! Try Again
OK
Rate Me
Report a Bug
Fibonacci Series
Prime numbers
Problem?
To find the Factorial of a Number.
-89
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

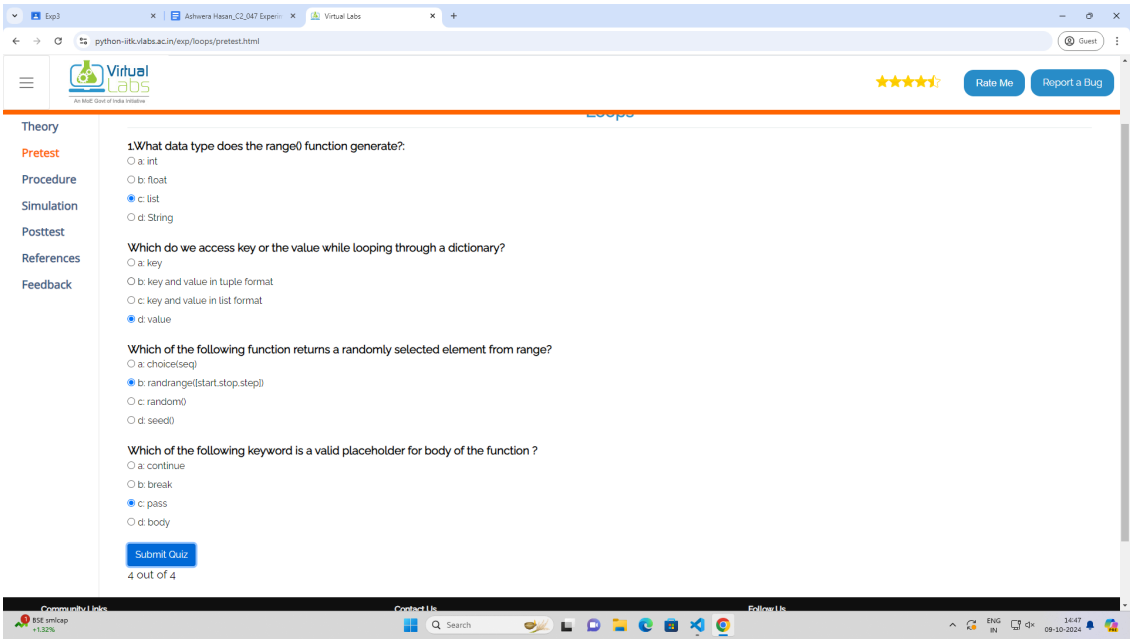
2.


Loops
Rate Me
Report a Bug
Analysis of Fibonacci Series
Back
Problem?
To find the Fibonacci Series.
10
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 :
0,1,1,2,3,5,8,13,21,34

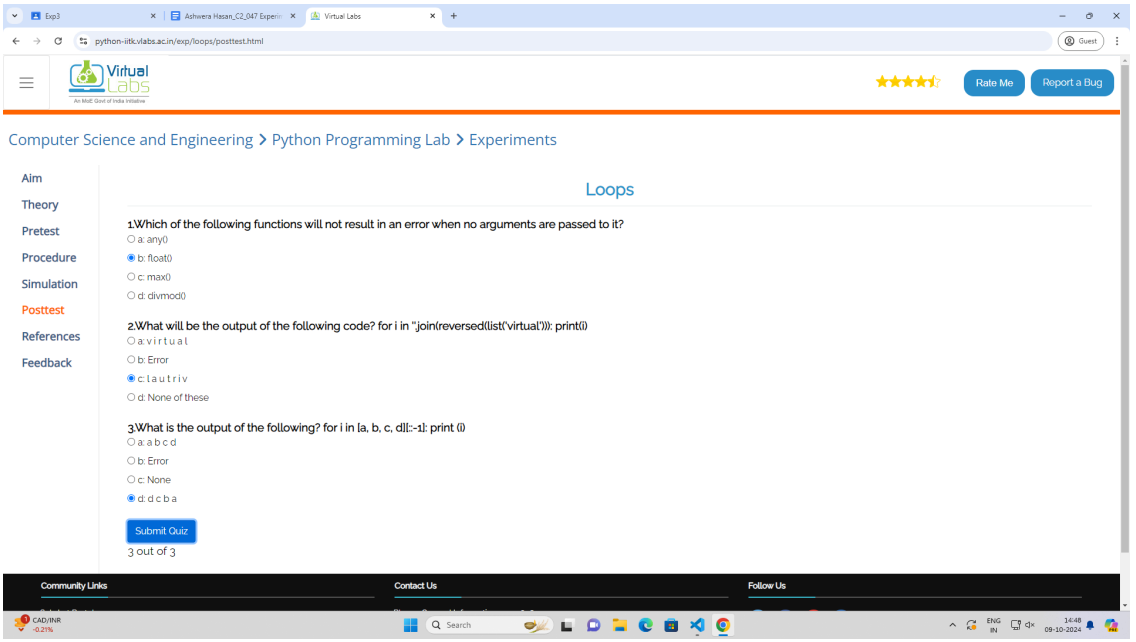

python-iitk.vlabs.ac.in says
Please enter a positive integer
OK
Rate Me
Report a Bug
Back
Problem?
To find the Fibonacci Series.
-23
Submit
Start
Reset
Step Execute
n1 = 0
n2 = 1
count = 0
if nterms <= 0:
print("Please enter a positive integer")
elif nterms == 1:
print("Fibonacci sequence upto",nterms,":")
print(n1)
else:
print("Fibonacci sequence upto",nterms,":")
while count < nterms:
Result



PreTest:

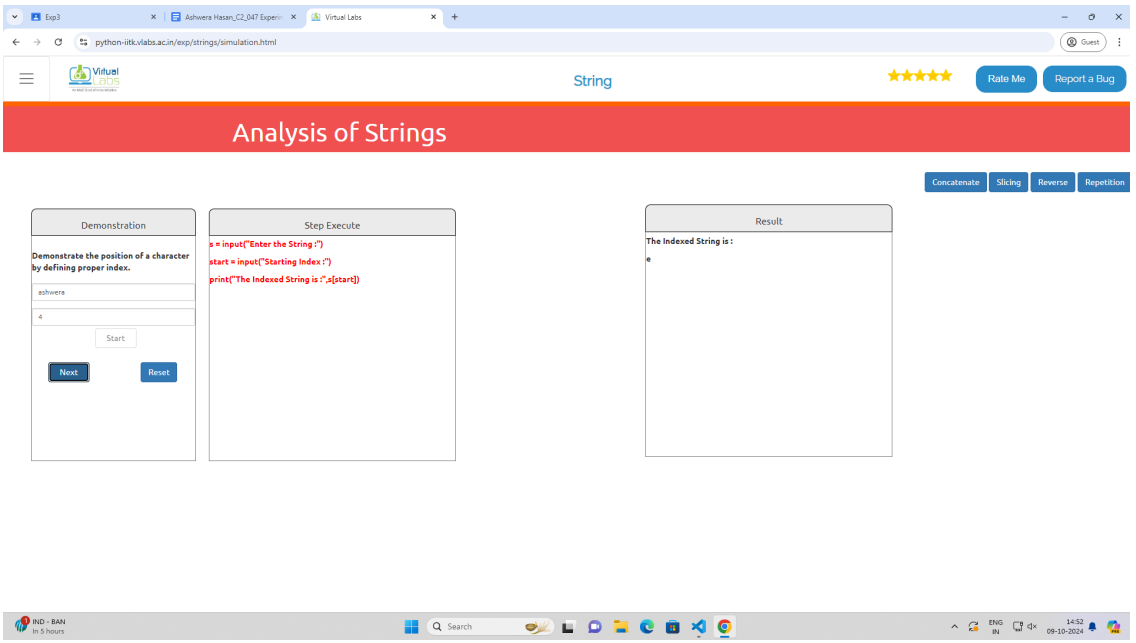


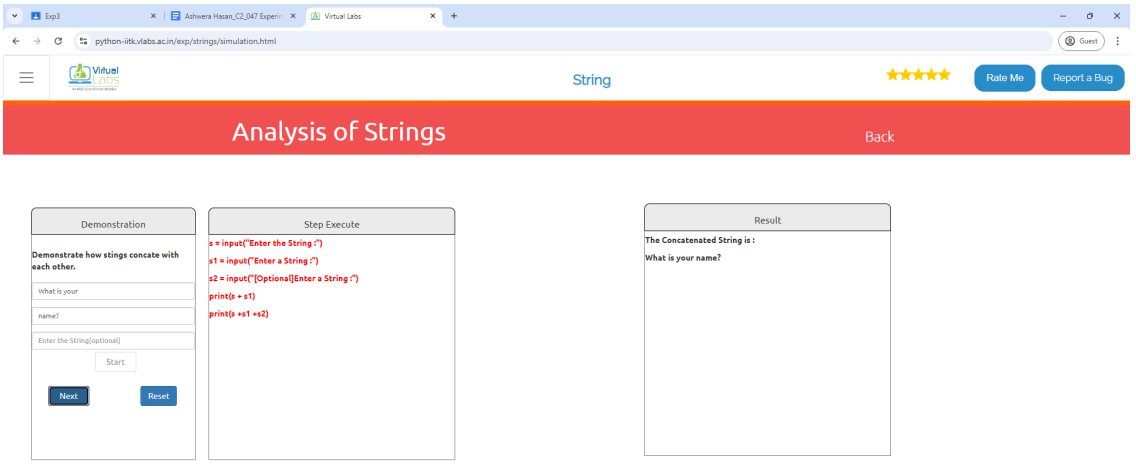
PostTest:

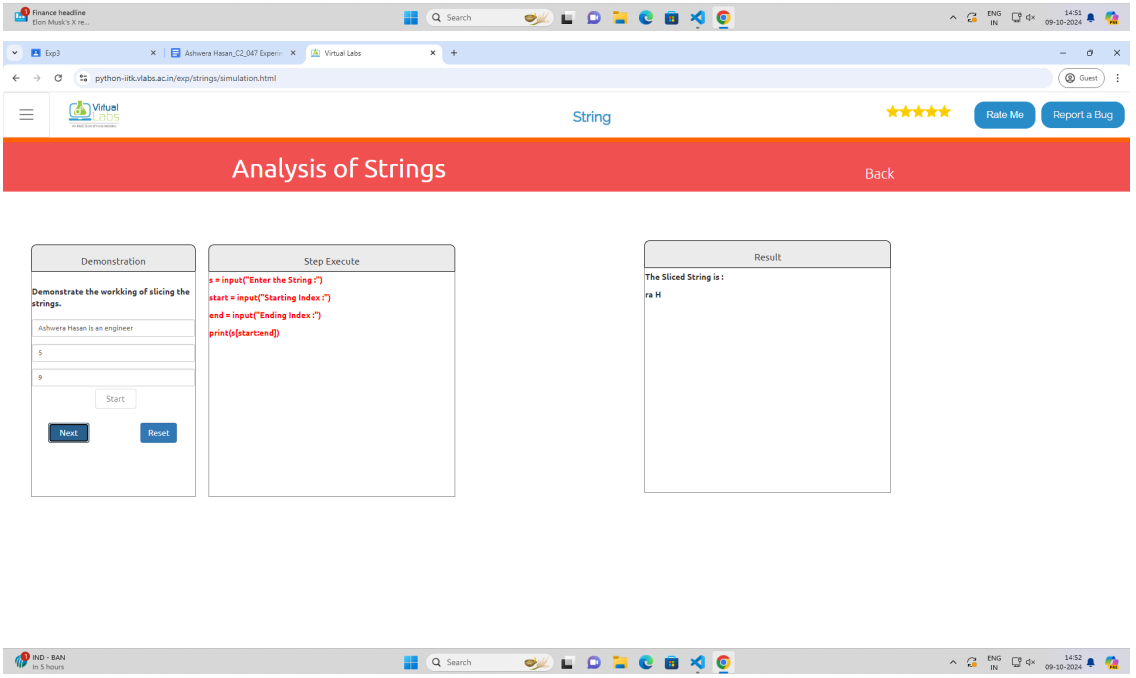


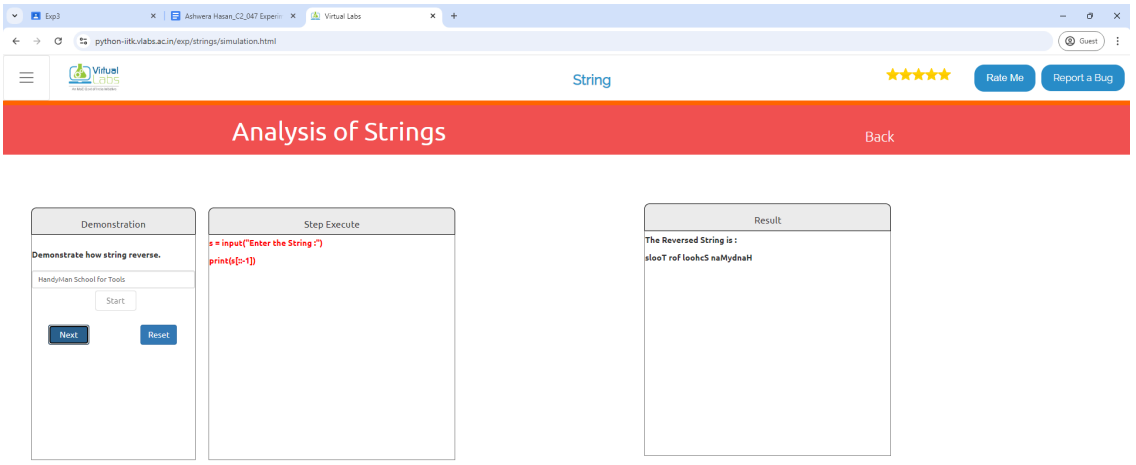
Virtual Lab Exp 2: Strings

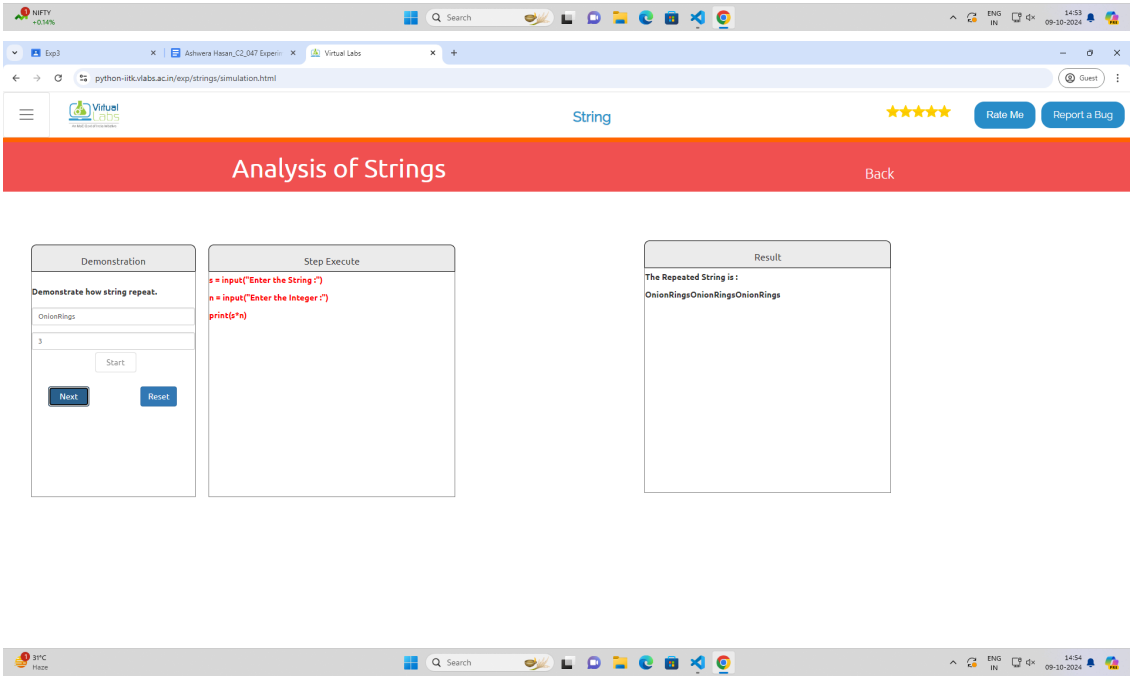
1.



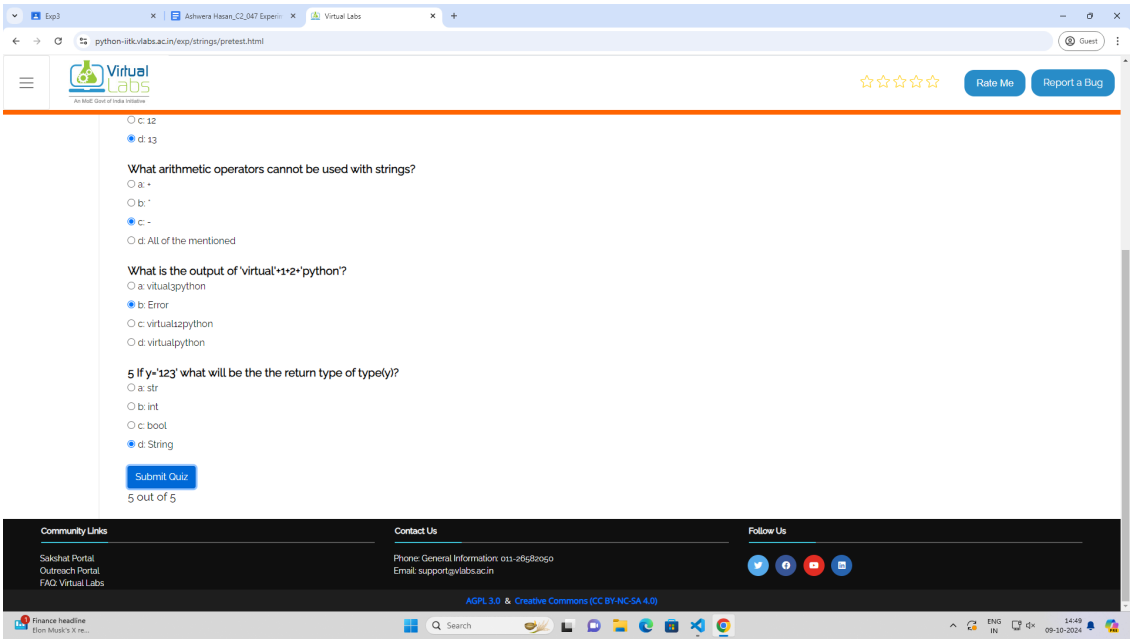

String
Rate Me
Report a Bug
Analysis of Strings
Back
Demonstration
Demonstrate how stings concate with each other.
What is your
name?
Enter the String[optional]
Start
Next
Reset
Step Execute
s = input("Enter the String :")
s1 = input("Enter a String :")
s2 = input("[Optional]Enter a String :")
print(s + s1)
print(s +s1 +s2)
Result
The Concatenated String is :
What is your name?


String
Rate Me
Report a Bug
Analysis of Strings
Back
Demonstration
Demonstrate the workking of slicing the strings.
Ashwera Hasan is an engineer
5
9
Start
Next
Reset
Step Execute
s = input("Enter the String :")
start = input("Starting Index :")
end = input("Ending Index :")
print(s[start:end])
Result
The Sliced String is :
ra H

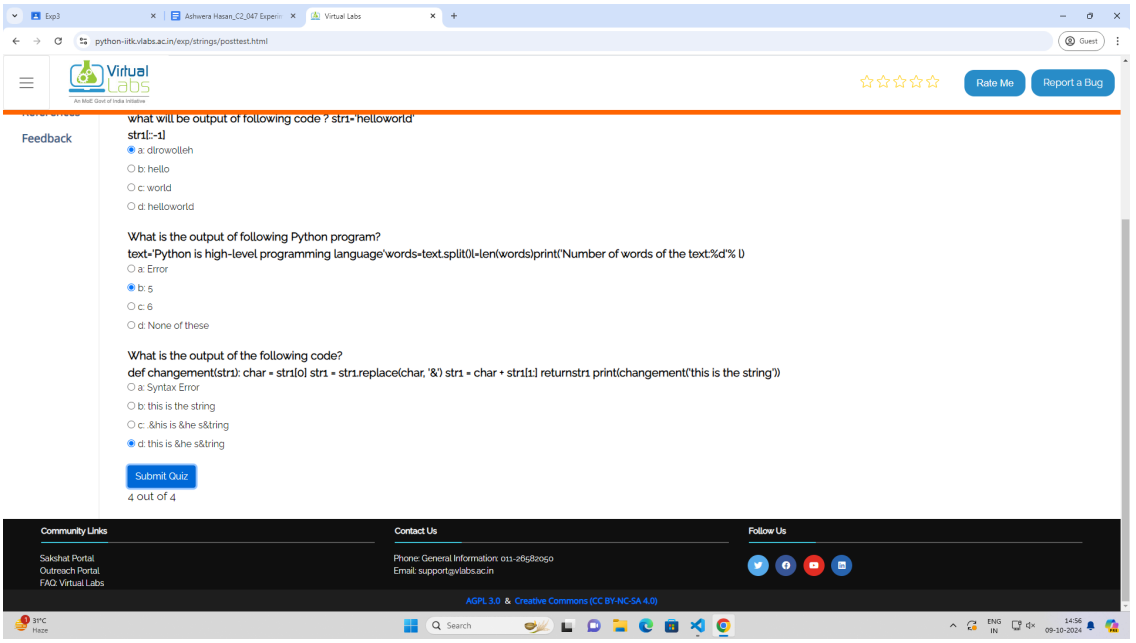

String
Rate Me
Report a Bug
Analysis of Strings
Back
Demonstration
Demonstrate how string reverse.
HandyMan School for Tools
Start
Next
Reset
Step Execute
s = input("Enter the String :")
print(s[::-1])
Result
The Reversed String is :
slooT rof loohcS naMydnaH


String
Rate Me
Report a Bug
Analysis of Strings
Back
Demonstration
Demonstrate how string repeat.
OnionRings
3
Start
Next
Reset
Step Execute
s = input("Enter the String :")
n = input("Enter the Integer :")
print(s*n)
Result
The Repeated String is :
OnionRingsOnionRingsOnionRings

PreTest


python-iitk.vlabs.ac.in/exp/strings/pretest.html
Virtual Labs
Rate Me
Report a Bug
c: 12
d: 13
What arithmetic operators cannot be used with strings?
a: +
b: *
c: -
d: All of the mentioned
What is the output of 'virtual'+1+2+'python'?
a: vitual3python
b: Error
c: virtual12python
d: virtualpython
5 If y='123' what will be the the return type of type(y)?
a: str
b: int
c: bool
d: String
Submit Quiz
5 out of 5
Community Links
Sakshat Portal
Outreach Portal
FAQ: Virtual Labs
Contact Us
Phone: General Information: 011-26582050
Email: support@vlabs.ac.in
Follow Us
AGPL 3.0 & Creative Commons (CC BY-NC-SA 4.0)

PostTest:


python-iitk.vlabs.ac.in/exp/strings/posttest.html
Virtual Labs
Rate Me
Report a Bug
Feedback
what will be output of following code ? str1='helloworld'
str1[::-1]
a: dlrowolleh
b: hello
c: world
d: helloworld
What is the output of following Python program?
text='Python is high-level programming language'words=text.split()l=len(words)print('Number of words of the text:%d'% l)
a: Error
b: 5
c: 6
d: None of these
What is the output of the following code?
def changement(str1): char = str1[0] str1 = str1.replace(char, '&') str1 = char + str1[1:] returnstr1 print(changement('this is the string'))
a: Syntax Error
b: this is the string
c: &his is &he s&tring
d: this is &he s&tring
Submit Quiz
4 out of 4
Community Links
Sakshat Portal
Outreach Portal
FAQ: Virtual Labs
Contact Us
Phone: General Information: 011-26582050
Email: support@vlabs.ac.in
Follow Us
AGPL 3.0 & Creative Commons (CC BY-NC-SA 4.0)