

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

Batch:C7-2 Roll No.:27
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")
```

Department of Department of Science and Humanities

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

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, **kwargs

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

Department of Department of Science and Humanities

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

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
<pre>def my_function(fname,lname): print(fname+ " " + lname) my_function("Amit", "Kumar")</pre>	Amit Kumar
<pre>def my_function(fname, lname): print(fname + " " + lname) my_function("Emil")</pre>	my_function() missing 1 required positional argument: 'lname'
<pre>def my_function(*kids): print("The youngest child is " + kids[2]) my_function("Emil", "Tobias", "Linus")</pre>	The youngest child is Linus
<pre>def my_function(college3, college2, college1): print("The Best college is " + college3) my_function("MIT", "Stanford", "Harvard")</pre>	The Best college is MIT
<pre>def my_function(country= "Norway"): print("I am from " + country) my_function("Sweden") my_function("India") my_function() my_function("Brazil")</pre>	I am from Sweden I am from India I am from Norway I am from Brazil
<pre>def tri_recursion(k):</pre>	1

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

if(k > 0):	3
result = k + tri_recursion(k - 1)	6
print(result)	10
else:	15
result = 0	21
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:

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.

```
def is_palindrome(str):
    if (len(str) <= 1):
        return True
    if (str[0] != str[-1]):
        return False
    return is_palindrome(str[1:-1])

str = str(input("enter a string "))

if (is_palindrome(str) == True):
    print(str, "is a palindrome")
else:
    print(str, "is not a palindrome")
```

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

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.

```
str=input("enter a string: ")
str=str.replace(" ", "")
dict={i:str.count(i) for i in set(str)}
for i in dict:
    x=dict[i]
    print(f"{i}:{x}")
```

Output(s):

3.

```
enter a string racecar
racecar is a palindrome
```

4.

```
enter a string: hello my name is nidhesh
e:3
a:1
l:2
i:2
s:2
y:1
o:1
n:2
m:2
h:3
d:1
```

Conclusion:

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

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

INPUT:

```
def factorial(n):
    if(n==0 or n==1):
        return 1
    else:
        return n*factorial(n-1)

n=int(input("Enter a positive number "))
print(f"The factorial of {n} is {factorial(n)}")
```

OUTPUT:

```
Enter a positive number 5
The factorial of 5 is 120
```

4. Define a function named 'test_range' that checks if a number 'n' is within the range 3 to 8 (inclusive)

INPUT:

```
def test_range(n):
    if (3<=n<=8):
        return True

n= int(input("Enter a number: "))
if(test_range(n)==True):
    print(f"{n} lies between 3 and 8")
else:
    print(f"{n} does not lie between 3 and 8")
```

OUTPUT:

```
Enter a number: 9
9 does not lie between 3 and 8
```

```
Enter a number: 5
5 lies between 3 and 8
```