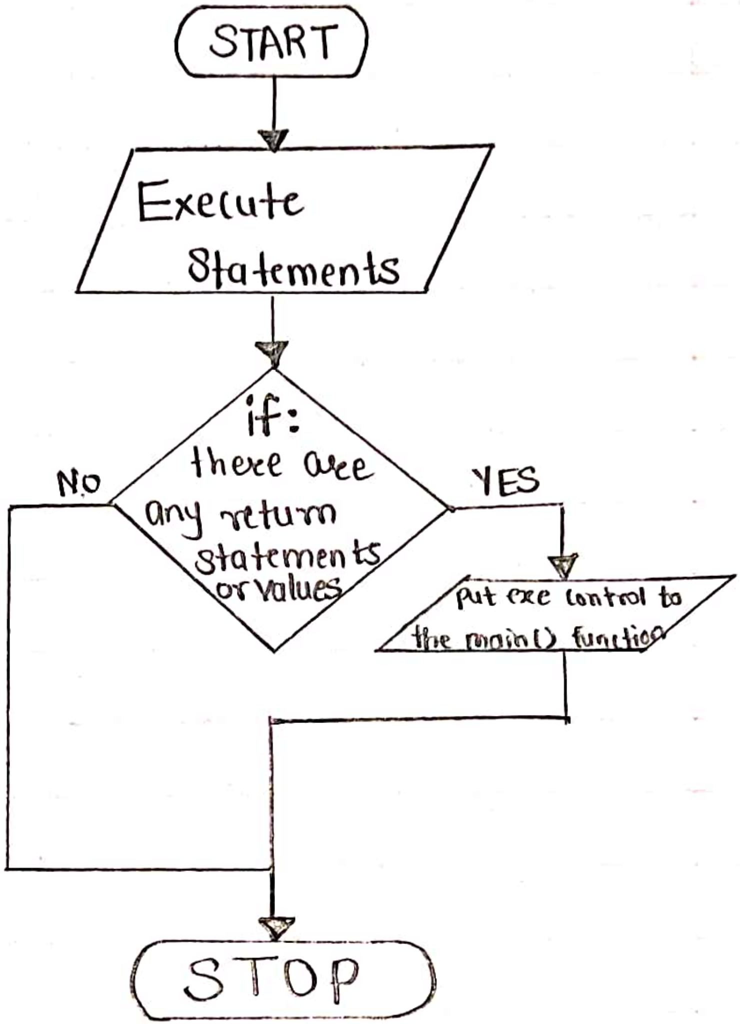
Practical No. 3

Date :

Aim : Implementing functions in python.

Flowchart:


START
Execute Statements
if: there are any return statements or values
No
YES
put exe control to the main() function
STOP

* Non-parametric functions *

Practical No. 3

Date:

Aim: Implementing functions in python

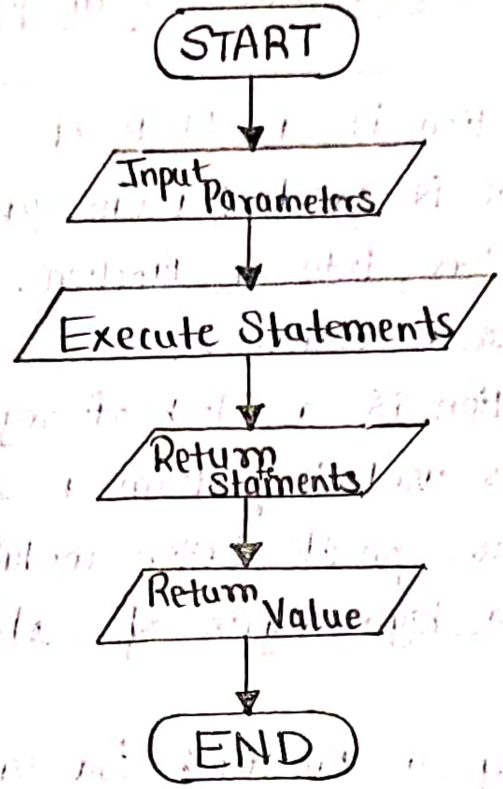
Theory: A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

Defining a function: You can define functions to provide the req. functionality. Here are simple rules to define a function in python.

- Function block begins with the keyword "def" followed by function name and parenthesis "()".
- Any inputs parameters or arguments should be placed within these parenthesis. You can also define parameters inside this parenthesis.
- The 1st element / statement of the function can be an optional statement - the documentation string of the function or "docstring".
- The code block within every function starts with a colon (:) & is indented.
- The statement returns a [expression] exsists a

Flowchart:-



* Parameterised Functions: *

function, optimally passing back an expression to the caller. A return statement with "no" arguments is same as return none.

Syntax :

```
def function_name (parameters):
        "function_doc_string"
        function_suite
        return [expression]
```

Theory : Parameters or Arguments.

The term parameter and argument can be used for the same thing : information that are passed into a function.

From a function's prespective :

A parameter is the variable listed inside the parenthesis in the function defination.

An argument is the value that is send to the function when it is called.

Number of Arguments :

By default, a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more & not less.

Conclusion: The experiment has been successfully executed.

Kernel: SageMath 10.4

CODE INDENTED BY RJD

Aim :- Write a Program in python to demonstrate the use of functions.

Declaring and calling a Normal Non Parameterized function.

```
In [6]:  def my_function():
             print("This is my Function.");


         my_function();
Out[6]: This is my Function.
```

```
In [7]:
         def my_function():
             print("Hello World.");
             print("This is my Function.");
         for i in range(4):
             my_function();
Out[7]: Hello World.
        This is my Function.
        Hello World.
        This is my Function.
        Hello World.
        This is my Function.
        Hello World.
        This is my Function.
```

Declaring and calling a Single Parameterized function.

```
In [8]:  def your_name(name):
             print("My Name is "+ name);

         your_name("Rushi");
Out[8]: My Name is Rushi
```

```
In [9]:  def your_name(name):
             print("My Name is "+ name);

         your_name("Rushi");
         your_name("Sumit");
         your_name("Shravan");
         your_name("Piyush");
         your_name("Harshit");
Out[9]: My Name is Rushi
        My Name is Sumit
        My Name is Shravan
        My Name is Piyush
        My Name is Harshit
```

Declaring and calling a Multi Parameterized function.

```
In [10]:  def your_name(name_1,name_2):
              print("My Name is " + name_1 + " and my name is "+ name_2 + "." + " We both Love each other");

          your_name("Meet","Shlok");
Out[10]: My Name is Meet and my name is Shlok. We both Love each other
```

```
In [11]:  def your_name(name_1,name_2):
              print("My Name is " + name_1 + " and my name is "+ name_2 + "." + " We both Love each other");

          your_name("Person_1","Person_2");
          your_name("Person_2","Person_1");
          your_name("Person_1","Person_2");
```

```
Out[11]: My Name is Person_1 and my name is Person_2. We both Love each other
         My Name is Person_2 and my name is Person_1. We both Love each other
         My Name is Person_1 and my name is Person_2. We both Love each other
```

Python Program to ADD two numbers using the concept of Parameter Passing in functions.

```
In [12]: def addTwoNumbers(num_1,num_2):
             num_3 = num_1 + num_2;
             print("The Numbers " + str(num_1) + " and " + str(num_2) + " Add to give " + str(num_3));

         addTwoNumbers(68,1);

Out[12]: The Numbers 68 and 1 Add to give 69
```

```
In [13]: def subTwoNumbers(num_1,num_2):
             num_3 = num_1 - num_2;
             print("The Numbers " + str(num_1) + " and " + str(num_2) + " Substract to give " + str(num_3));

         subTwoNumbers(70,1);

Out[13]: The Numbers 70 and 1 Substract to give 69
```

Printing a String List Using a Function.

```
In [14]: def myList():
             for x in fruits:
                 print(x);
         fruits = ["Banana","Apple","Gauva","Mango"];

         myList();

Out[14]: Banana
         Apple
         Gauva
         Mango
```

Printing a Number List Using a Function.

```
In [15]: def myList():
             for x in fruits:
                 print(x);
         fruits = [1,2,3,4,5,6,7,8,9,10];

         myList();

Out[15]: 1
         2
         3
         4
         5
         6
         7
         8
         9
         10
```

Python program to print Factorial of a Number.

```
In [18]: def factorial(num):
             result = 1
             for i in range(1, num + 1):
                 result *= i
             print("The factorial of " + str(num) + " is " + str(result))

         factorial(1);
         factorial(3);
         factorial(5);
         factorial(10);

Out[18]: The factorial of 1 is 1
         The factorial of 3 is 6
         The factorial of 5 is 120
         The factorial of 10 is 3628800
```

Conclusion :- The Experiment has been successfully Executed.

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 : initialized.

Passing a List as an Argument :

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

Eg: if you send a List as an argument, it will still be a List when it reaches the function.

Return Values :

To let a function return a value, use the keyword "return value-to be returned" statement.

Conclusion :- The experiment has been successfully executed.