FUNCTIONS
VIVA QUESTIONS

1. **What is a function?**

   Function is a named group of related programming statements which perform a specific task. Functions are used to break the large code into smaller manageable modules which are easy to manage and debug.

2. **Need of functions**

   Some of the major advantages of using functions in a program are:

   - **Modularity**: Functions provide modularity to a program. It is difficult to manage large programs. Thus, a large program is usually broken down into small units called functions. This makes the program easy to understand, more organized and manageable.
   - **Code Reusability**: Functions **save space and time**. Sometime a part of the program code needs to be executed repeatedly. If this part of program code is stored in a function than we do not need to write the same code again and again. Instead, function can be called repeatedly. This saves time and space. We can also store the frequently used functions in the form of modules which can easily be imported in other programs when needed.
   - **Easy debugging**: It is easy to debug small units of a program than a long program.

3. **Types of functions**

   Basically, we can divide functions into the following three types:

   - Built-in Functions
   - Python Modules
   - User-defined Functions

4. **What is the difference between global and local variables?**
   A variable declared outside the function is known as ***global variable***. The global variable can be accessed inside or outside of the function.
   A variable declared inside the function is known as ***local variable***. The local variable can be accessed only inside the function in which it is declared.

5. **How can we access a global variable inside a function if a local variable exists inside the function by the same name?**

   The keyword global may be used to access a global variable inside a function if a local variable exists inside the function by the same name.

6. **Consider the following set of statements:**

Program 1	Program 2
x=9 def func():	x=9 def func():

x=10 print("x=",x) func() print(x)	global x x=10 print("x=",x) func() print(x)

What will be the output of the above programs and why?

The output will be:

Program 1	Program 2
x= 10 9	x= 10 10

This is because in Python, variables that are only referenced inside a function are implicitly global. If a variable is assigned a value or its value is modified anywhere within the function's body, it's assumed to be a local unless explicitly declared as global.

7. **What is the difference between parameters and arguments?**

Information can be passed to functions using parameters. They are specified inside the parenthesis after the function name and are separated by a comma.
The values provided in the function call are called arguments.
Arguments and parameters have one to one correspondence.

8. **What is the difference between Positional arguments and Keyword/Named arguments?**

When arguments are passed as positional arguments, the order in which the arguments are sent or their position is very important.

- It is mandatory to pass arguments in the same order as the parameters.
- The arguments should have one to one correspondence/mapping with parameters present in the function header.
- The number and type of arguments passed should match the number and type of parameters present in the function header.

When arguments are passed as keyword arguments, the name of the argument is given along with the value. In this case the position of arguments does not matter. They can be written in random order.

9. **What is wrong in the following function call?**
test(a = 1, b = 2, c = 3, 4) #3

Positional arguments cannot follow keyword arguments.

10. What is the purpose of a return statement? How can a function return more than one value?

The ***return*** statement is used to return either a single value or multiple values from a function. More than values are returned by a function in the form of a tuple.