A logo with blue and red text

Description automatically generated A logo with text on it

Description automatically generated

**Introduction to Programming**

**Lab Worksheet**

**Week 5**

**Python**

Name: Nikita Sah

Level 4 Section: A

British Id: 10011

LBU Id: c7576150

Level 4 BSc. Hons Computing

Subject: Fundamental of Computer Programming

(FOCP)

The British College (TBC)

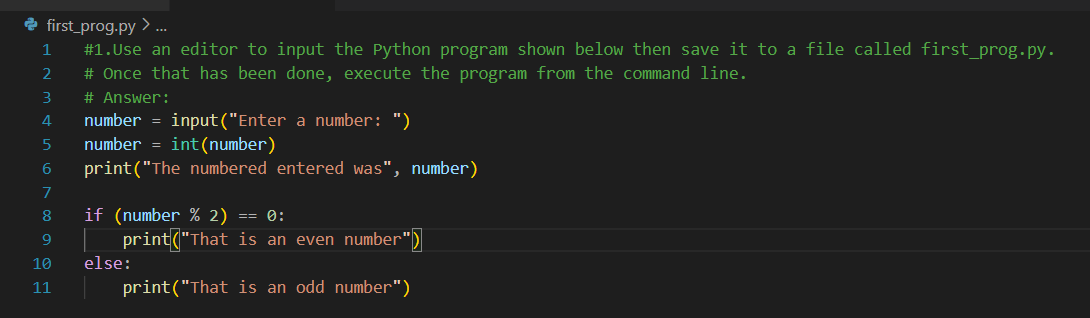
**Task:**

**#Week 5 Practical**

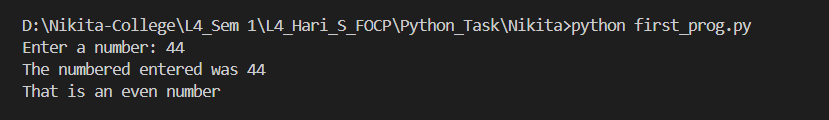
1. Use an editor to input the Python program shown below then save it to a file called first\_prog.py. Once that has been done, execute the program from the command line.

**Answer:**

**Source code of Question No. 1:**

****

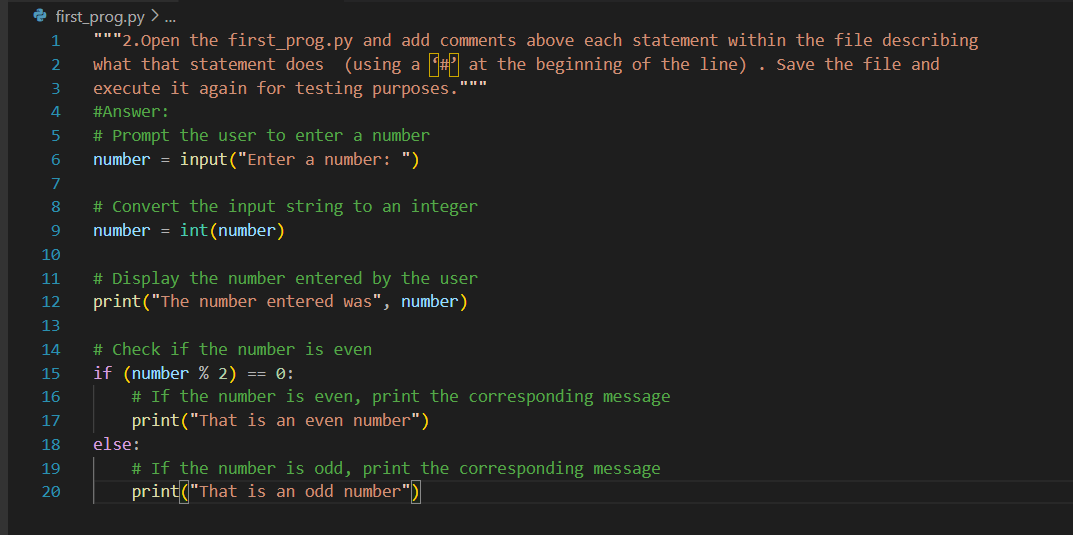
**Output of Question No. 1:**

****

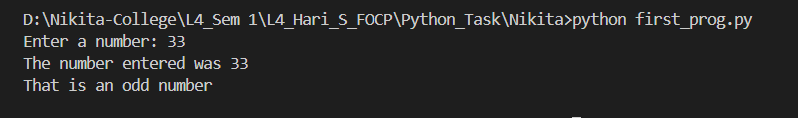
1. Open the first\_prog.py and add comments above each statement within the file describing what that statement does (using a ‘#’ at the beginning of the line) . Save the file and execute it again for testing purposes.

**Answer:**

**Source code of Question No. 2:**

****

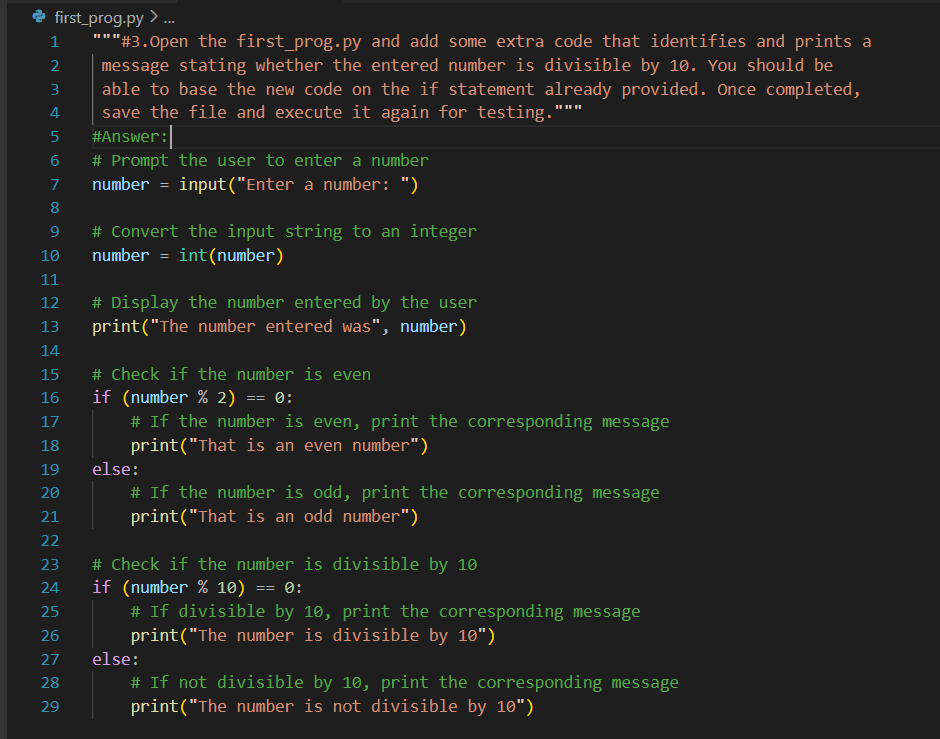
**Output of Question No. 2:**

****

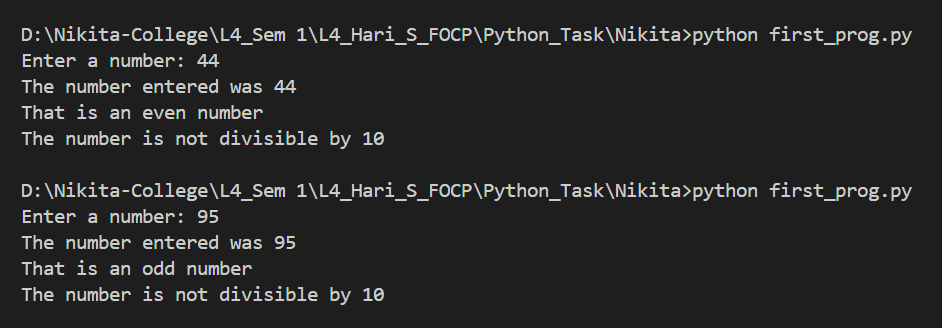
1. Open the first\_prog.py and add some extra code that identifies and prints a message stating whether the entered number is divisible by 10. You should be able to base the new code on the if statement already provided. Once completed, save the file and execute it again for testing.

**Answer:**

**Source code of Question No. 3:**

****

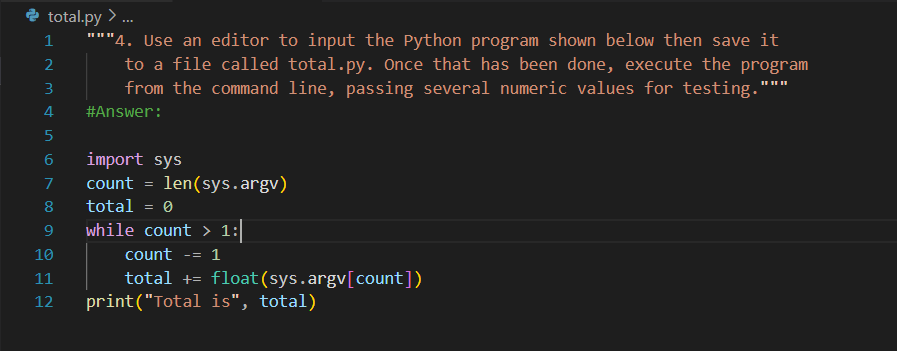
**Output of Question No. 3:**



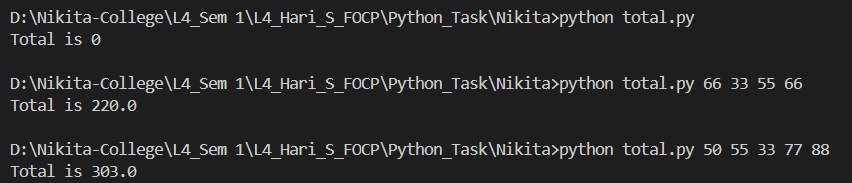
1. Use an editor to input the Python program shown below then save it to a file called total.py. Once that has been done, execute the program from the command line, passing several numeric values for testing.

**Answer:**

**Source code of Question No. 4:**

****

**Output of Question No. 4:**

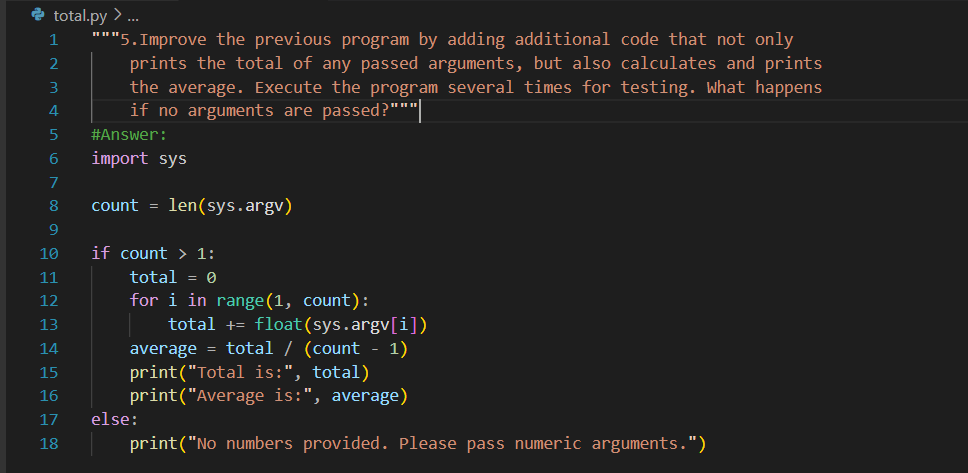


1. Improve the previous program by adding additional code that not only prints the total of any passed arguments, but also calculates and prints the average. Execute the program several times for testing. What happens if no arguments are passed?

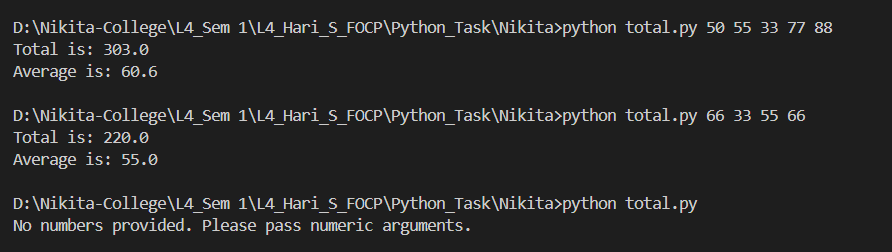
#Remember, the first element of the sys.argv list is the program name itself, so it is not uncommon to ignore or skip processing of that first element.

**Answer:**

**Source code of Question No. 5:**

****

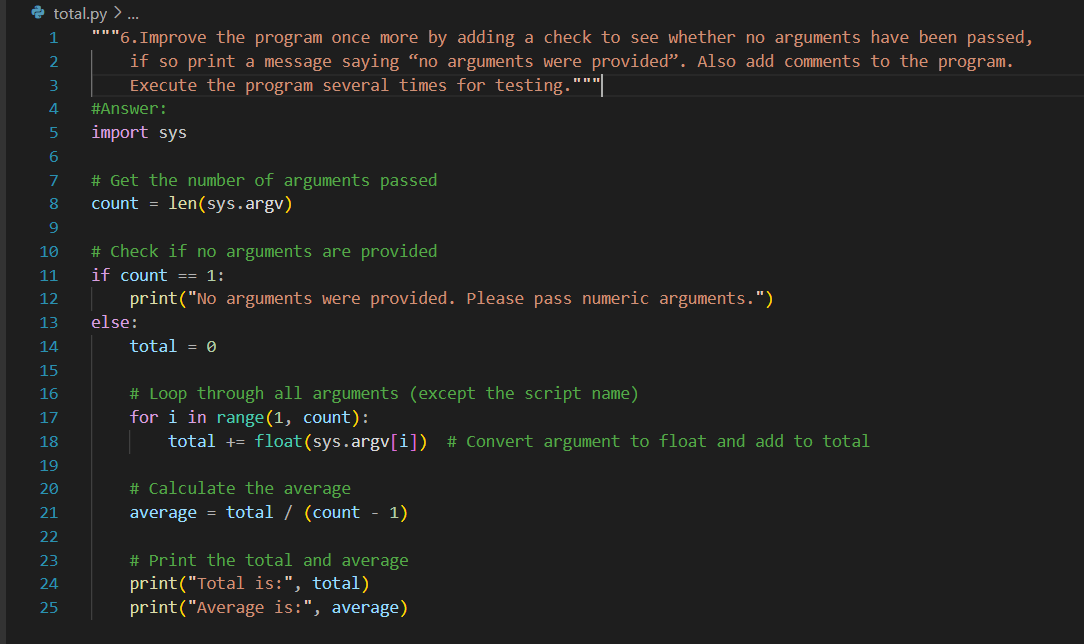
**Output of Question No. 5:**

****

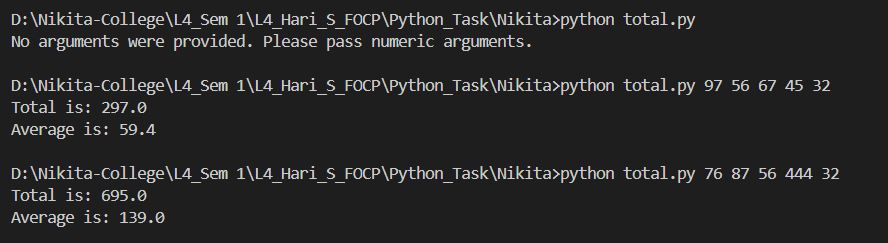
1. Improve the program once more by adding a check to see whether no arguments have been passed, if so print a message saying “no arguments were provided”. Also add comments to the program. Execute the program several times for testing.

**Answer:**

**Source code of Question No. 6:**

****

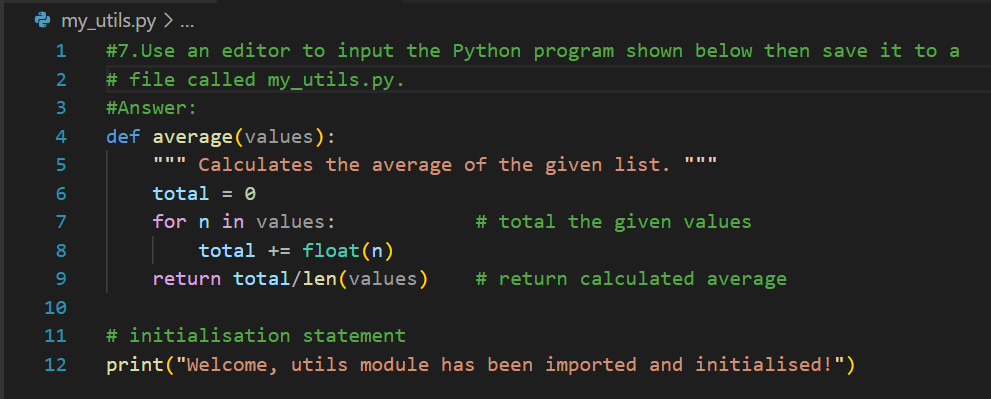
**Output of Question No. 6:**



1. Use an editor to input the Python program shown below then save it to a file called my\_utils.py.

**Answer:**

**Source code of Question No. 7:**

****

**Output of Question No. 7:**

A black background with white text

Description automatically generated

1. Use an editor to input another Python program utils\_test.py. This program should import my\_utils then call the average() function several times, passing a list of values as a parameter, e.g.

print("Average is", my\_utils.average([10, 23, 30]))

print("Another average is", my\_utils.average([10.2, 8.8, 2.6]))

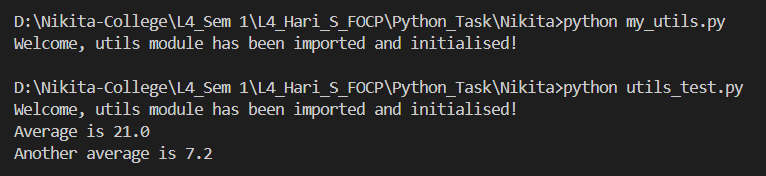
**Answer:**

**Source code of Question No. 8:**

**A screen shot of a computer program

Description automatically generated**

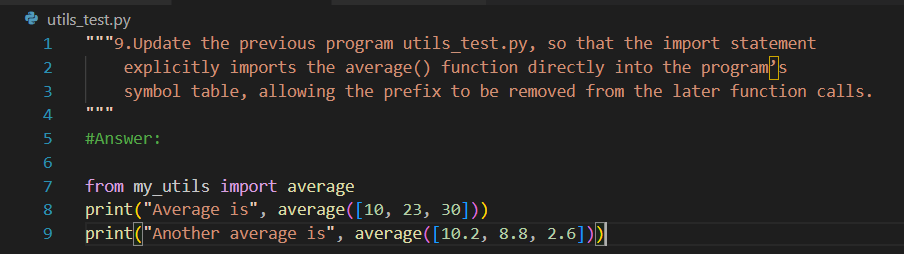
**Output of Question No. 8:**

****

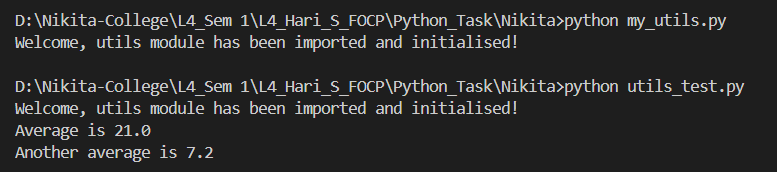
1. Update the previous program utils\_test.py, so that the import statement explicitly imports the average() function directly into the program’s *symbol table*, allowing the prefix to be removed from the later function calls.

**Answer:**

**Source code of Question No. 9:**

****

**Output of Question No. 9:**



1. Start Python in interactive mode and input the following statements.

import math

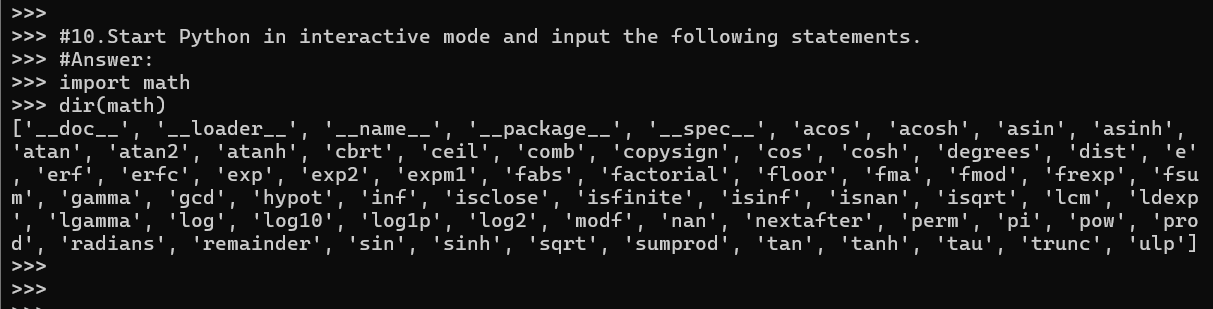
dir(math)

**Answer:**

import math

dir(math)

**Output of Question No. 10:**



1. Now enter the following statements:

import math

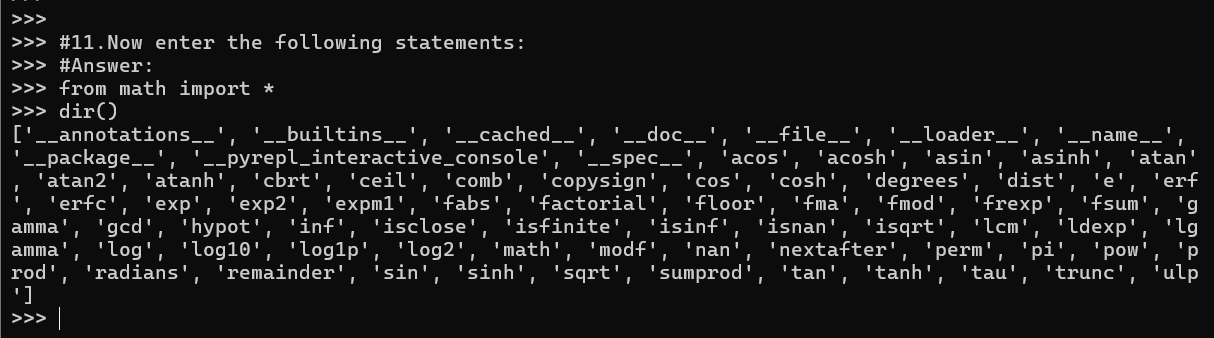
dir(math)

**Answer:**

import math

dir(math)

**Output of Question No. 11:**



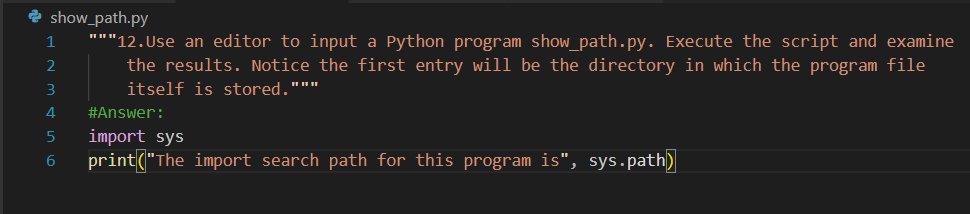
1. Use an editor to input a Python program show\_path.py. Execute the script and examine the results. Notice the first entry will be the directory in which the program file itself is stored.

import sys

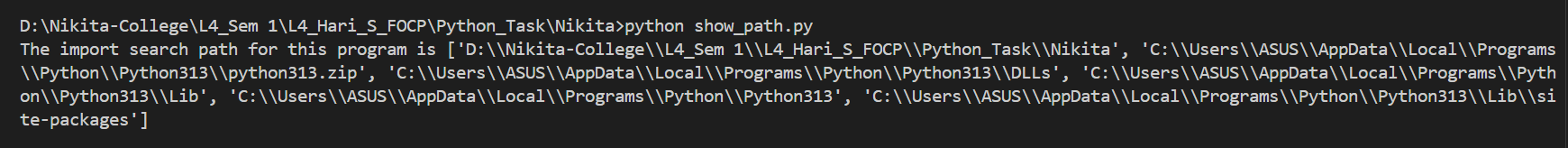
print(“The import search path for this program is ”,sys.path)

**Answer:**

**Source code of Question No. 12:**

****

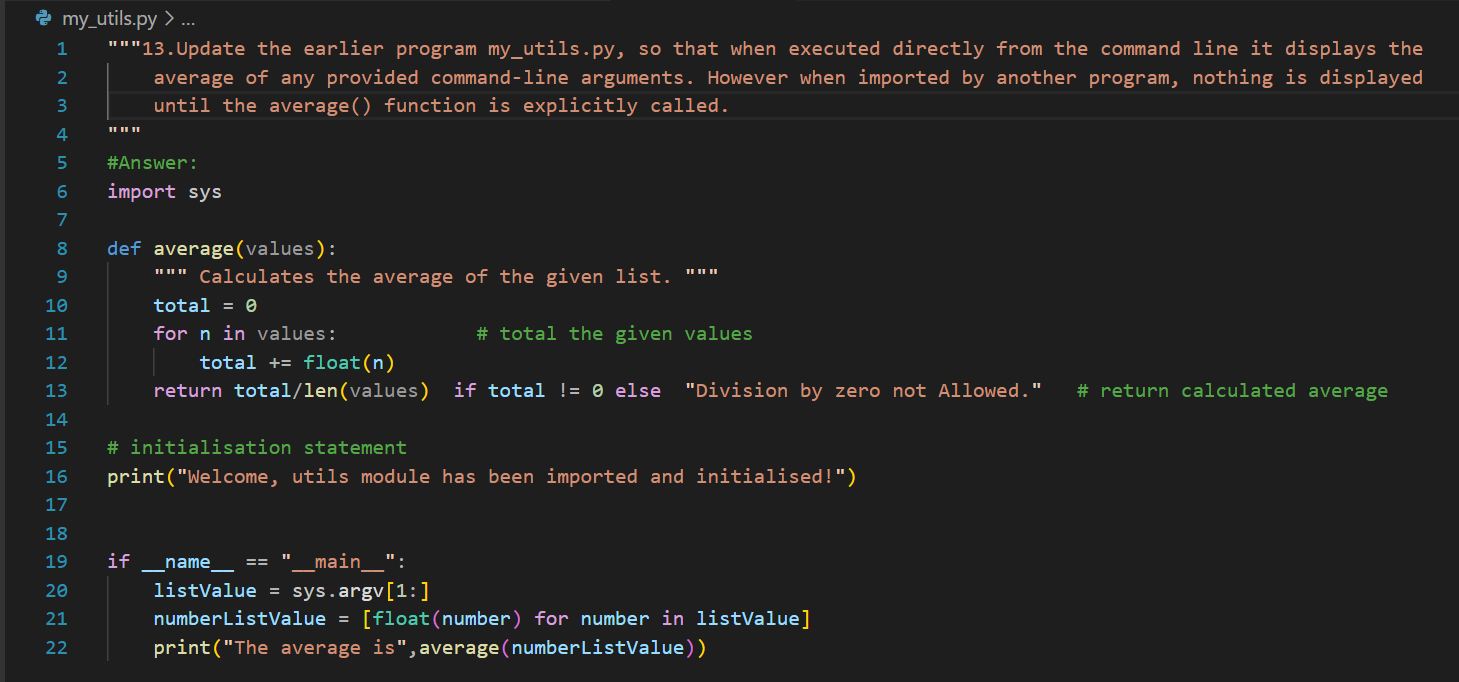
**Output of Question No. 12:**



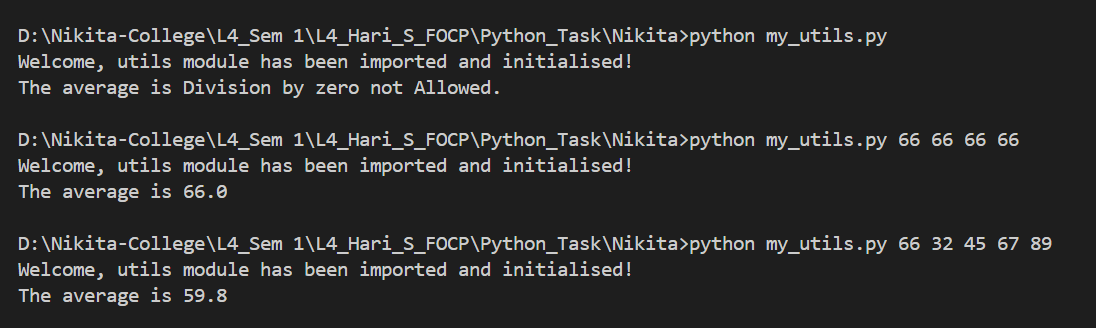
1. Update the earlier program my\_utils.py, so that when executed directly from the command line it displays the average of any provided *command-line arguments*. However, when imported by another program, nothing is displayed until the average() function is explicitly called.

**Answer:**

**Source code of Question No. 13:**

****

**Output of Question No.13:**



1. Look at each of the phrases below and ensure you understand what each of these means. For any that you do not understand, do a little research to find a definition of each term. This research may involve looking back over these notes, or the associated lecture notes. It may also involve searching for these terms on the Internet.

* IDE
* Module
* Command Line Arguments
* Symbol-table
* Search path

**Answer:**

* **IDE:** An IDE (Integrated Development Environment) is software that combines commonly used developer tools into a compact GUI (graphical user interface) application.
* **Module:** Module is a file containing Python code that can define functions, classes, and variables, and can also include runnable code  that can be imported inside another Python Modules Operations Program.
* **Command Line Arguments:** The arguments that are given after the name of the program in the command line shell of the operating system are known as Command Line Arguments.
* **Symbol-table:** A symbol table is a data structure, where all the identifiers used in a program are stored along with their type, scope, and memory locations.
* **Search path:** The directory from which the input script was run, or the current directory if the interpreter is being run interactively.