डॉ. ए.पी.जे. अब्दुल कलाम प्रौद्योगिकी संस्थान, टनकपुर, उत्तराखंड
(संघटक संस्थान, उत्तराखंड तकनीकी विश्वविद्यालय, उत्तराखंड सरकार)
Dr. A.P.J. Abdul Kalam Institute of Technology, Tanakpur, Uttarakhand
(A Constituent Institute of Uttarakhand Technical University, Govt. of Uttarakhand)
Approved by AICTE, MHRD New Delhi

LAB FILE

Academic Session
(2022-23)

Programme	: B. Tech in Computer Science and Engineering
Course	: Computer Workshop (Python Programming)
No. of credits	: 1
Semester	: III
Session	: July-Dec 2022
Batch	: 2021-2025
Faculty	: Dr. Hitesh Kumar Sharma

Submitted By:

Roll No: 211620101001

Name: AAYUSH BIJALWAN

Submitted to:

Dr. Hitesh Kumar Sharma

Dept. of Computer Science

LIST OF EXPERIMENTS

Exp No.	Experiment Name	Date of Experiment
1	How to declare and use variables and operators	
2	Programming using Basic Libraries (Numpy, Pandas, SK Learn etc)	
3	To write a Python program to print HELLO INDIA.	
4	To write a Python program that takes in command line arguments as input and print the number of arguments.	
5	To write a Python program find the division of student.	
6	To write a program implements Fibonacci series.	
7	To write a Python program for factorial.	
8	To write a Python program to use of functions.	
9	To write a Python program to implement list.	
10	To write a Python program to implement tuples.	
11	To write a Python program Insertion sort.	
12	To write a Python program merge sort.	
13	To write a Python program first n prime numbers.	
14	Implementation of Data Science concepts using Python	

LAB EXERCISE 1

Aim:-How to declare and use variables and operators

DESCRIPTION:-Declaring variables

One of the most powerful features of a programming language is the ability to define and manipulate **variables**. A variable is a named location that stores a **value**. Values may be numbers, text, images, sounds, and other types of data. To store a value, you first have to declare a variable.

```
String message;
```

This statement is a **declaration**, because it declares that the variable named `message` has the type `String`. Each variable has a **type** that determines what kind of values it can store. For example, the `int` type can store integers, and the `char` type can store characters.

Some types begin with a capital letter and some with lowercase. We will learn the significance of this distinction later, but for now you should take care to get it right. There is no such type as `Int` or `string`.

To declare an integer variable named `x`, you simply type:

```
int x;
```

Note that `x` is an arbitrary name for the variable. In general, you should use names that indicate what the variables mean. For example, if you saw these declarations, you could probably guess what values would be stored:

```
String firstName;
String lastName;
int hour, minute;
```

This example declares two variables with type `String` and two with type `int`. When a variable name contains more than one word, like `firstName`, it is conventional to capitalize the first letter of each word except the first. Variable names are case-sensitive, so `firstName` is not the same as `firstname` or `FirstName`.

This example also demonstrates the **syntax** for declaring multiple variables with the same type on one line: `hour` and `minute` are both integers. Note that each declaration statement ends with a semicolon.

You can use any name you want for a variable. But there are about 50 reserved words, called **keywords**, that you are not allowed to use as variable names. These words include `public`, `class`, `static`, `void`, and `int`, which are used by the compiler to analyze the structure of the program.

2.2 Assignment

Now that we have declared variables, we want to use them to store values. We do that with an **assignment** statement.

```
message = "Hello!";  // give message the value "Hello!"

hour = 11;           // assign the value 11 to hour

minute = 59;         // set minute to 59
```

This example shows three assignments, and the comments illustrate different ways people sometimes talk about assignment statements. The vocabulary can be confusing here, but the idea is straightforward:

- When you declare a variable, you create a named storage location.
- When you make an assignment to a variable, you update its value.

As a general rule, a variable has to have the same type as the value you assign to it. For example, you cannot store a string in `minute` or an integer in `message`. We will see some examples that seem to break this rule, but we'll get to that later.

A common source of confusion is that some strings *look* like integers, but they are not. For example, `message` can contain the string `"123"`, which is made up of the characters `'1'`, `'2'`, and `'3'`. But that is not the same thing as the integer `123`.

```
message = "123";     // legal
```

```
message = 123;        // not legal
```

Variables must be **initialized** (assigned for the first time) before they can be used. You can declare a variable and then assign a value later, as in the previous example. You can also declare and initialize on the same line:

```
String message = "Hello!";

int hour = 11;

int minute = 59;
```

LAB EXERCISE 2

Aim:-Programming using Basic Libraries (Numpy, Pandas, SK Learn etc)

DESCRIPTION:- Till today I have written all tutorials without libraries and now I'm taking our journey to next level where we will use python libraries for classification, visualization and clustering. In this article, we will have a short introduction of NumPy, SciPy, matplotlib, scikit-learn, pandas.

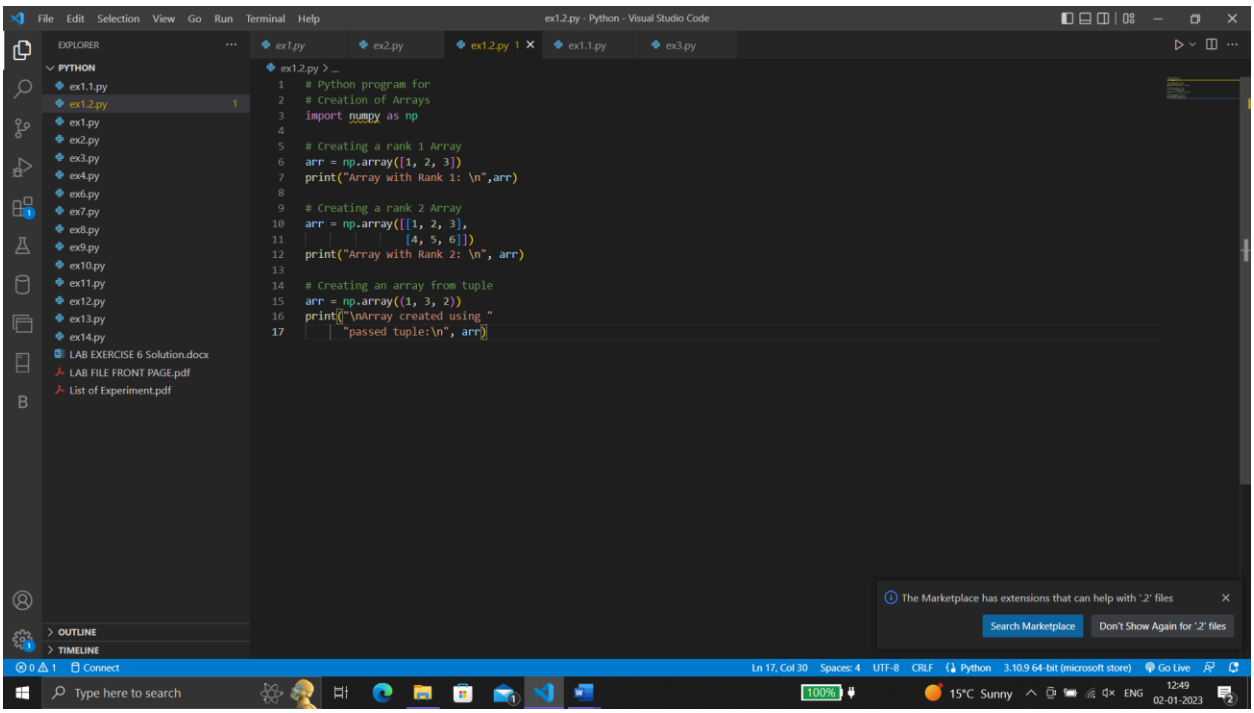
NumPy

NumPy basically provides n-dimensional array object. NumPy also provides mathematical functions which can be used in many calculations.

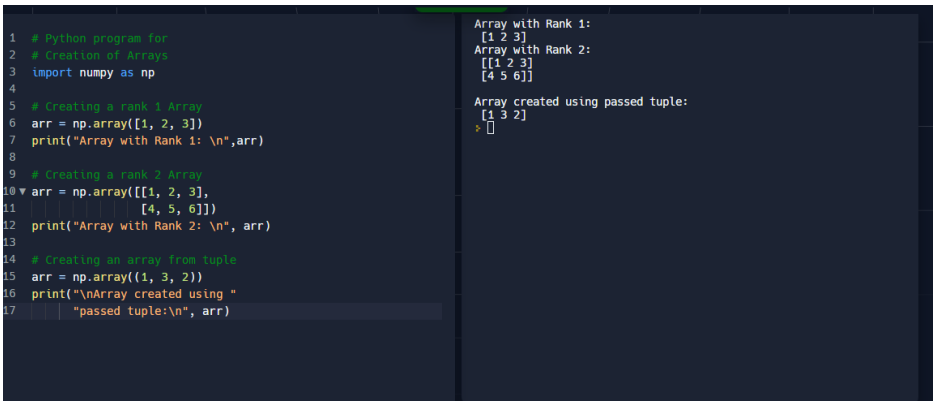
Command to install: *pip install numpy*

SOLUTION:-

CODE:--



OUTPUT:-



```python
# Python program for
# Creation of Arrays
import numpy as np

# Creating a rank 1 Array
arr = np.array([1, 2, 3])
print("Array with Rank 1: \n",arr)

# Creating a rank 2 Array
arr = np.array([[1, 2, 3],
                [4, 5, 6]])
print("Array with Rank 2: \n", arr)

# Creating an array from tuple
arr = np.array((1, 3, 2))
print("\nArray created using "
      "passed tuple:\n", arr)
```

```
Array with Rank 1:
 [1 2 3]
Array with Rank 2:
 [[1 2 3]
 [4 5 6]]

Array created using passed tuple:
 [1 3 2]
```

LAB EXERCISE 3

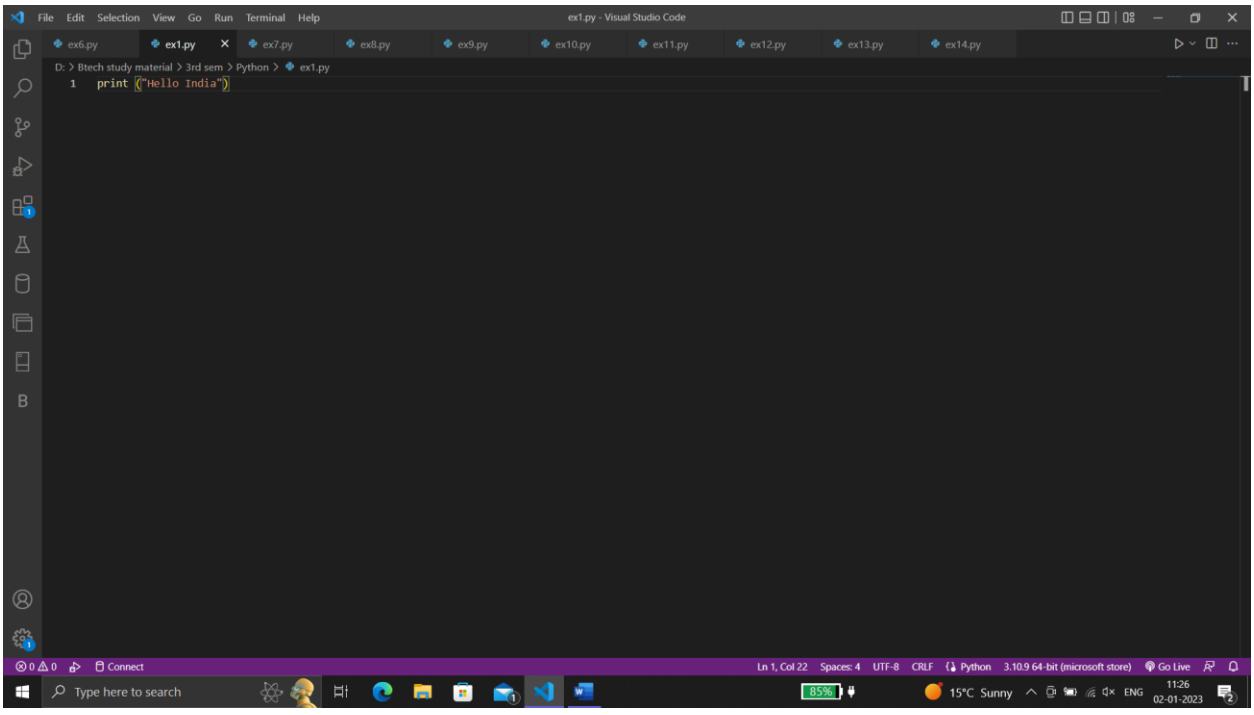
Aim: To write a Python program to print HELLO INDIA.

Description:

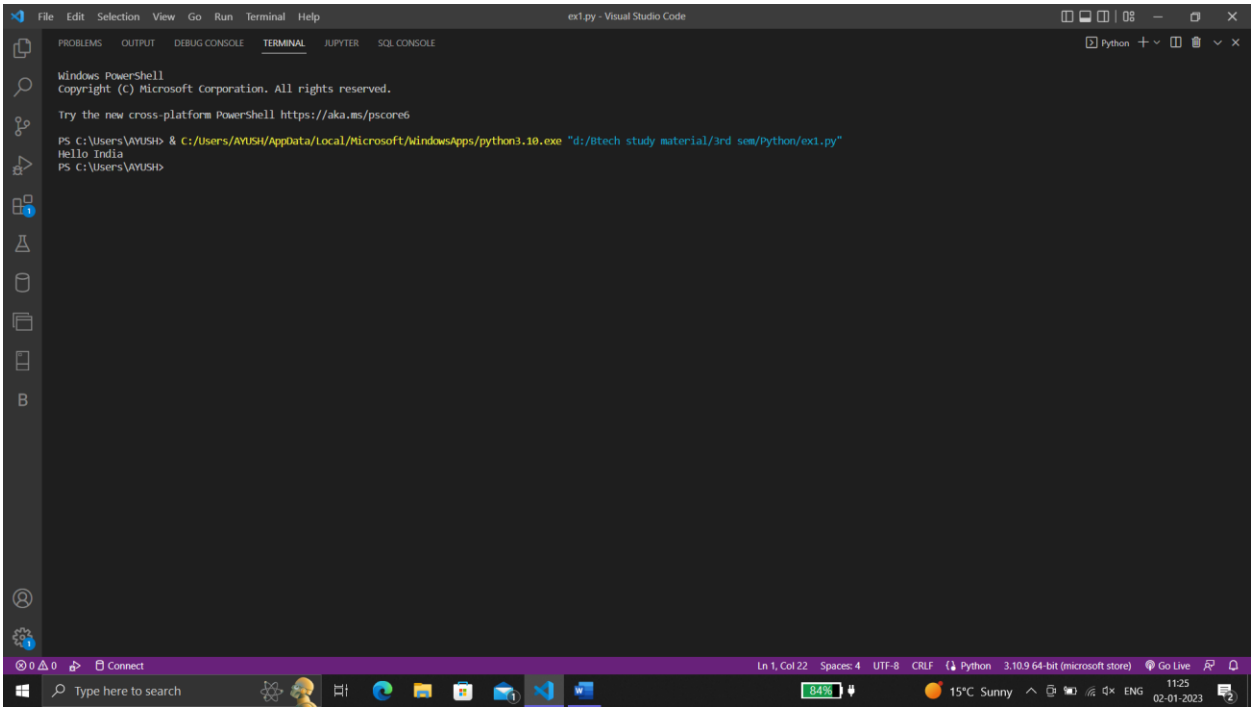
In this program, we have used the built-in function to print the string HELLO INDIA on our screen.

Solution:

Code:



Input/Output:



LAB EXERCISE 4

Aim: To write a Python program that takes in command line arguments as input and print the number of arguments.

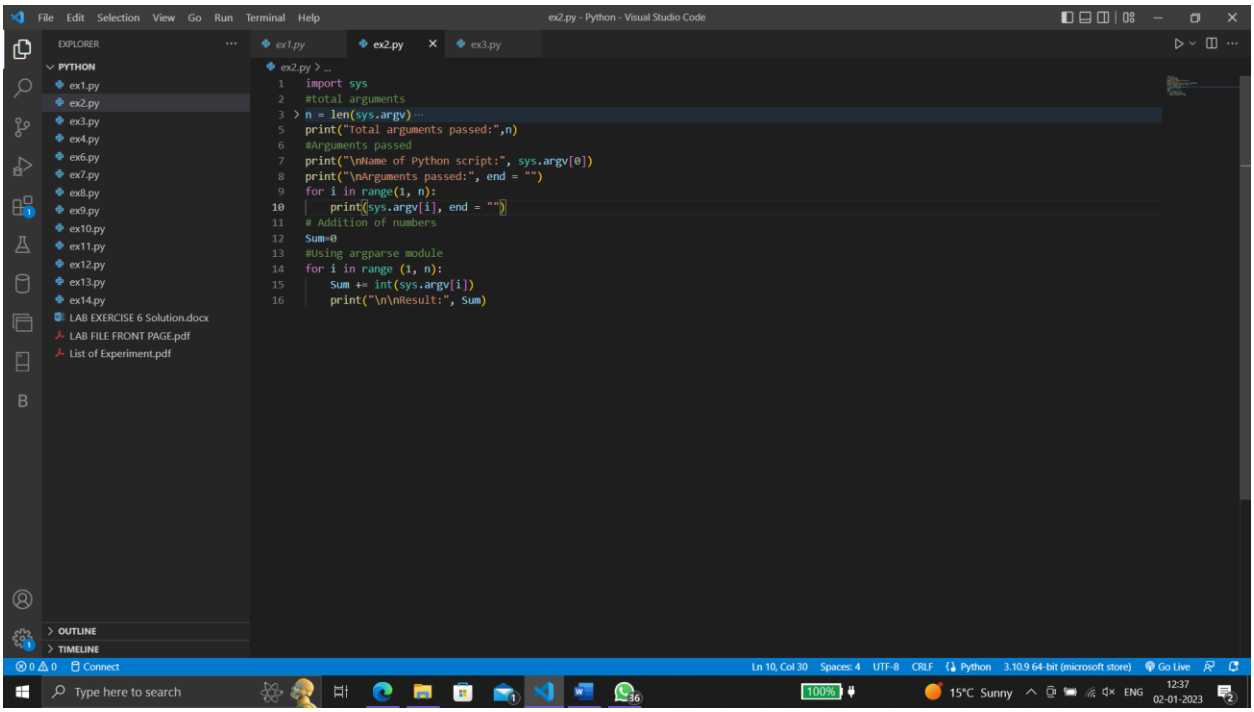
Description:

The argument that are given after the name of program in the command line shell of the operating system are known as
Command Line Arguments.
One such variable is sys.argv which is a simple list structure. It's main purpose are:

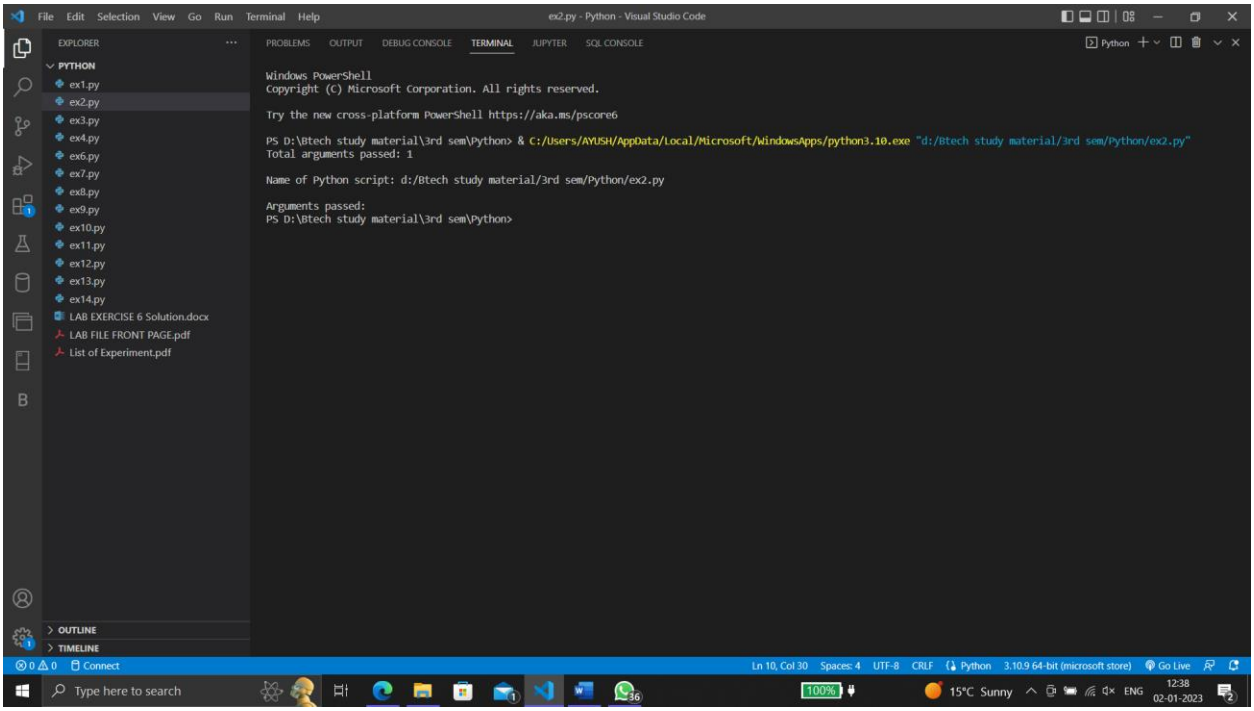
- It is a list of command line arguments.
- len(sys.argv) provides the number of command line arguments.
- sys.argv[0] is the name of the current Python script.

Solution:

Code:



Input/Output:



LAB EXERCISE 5

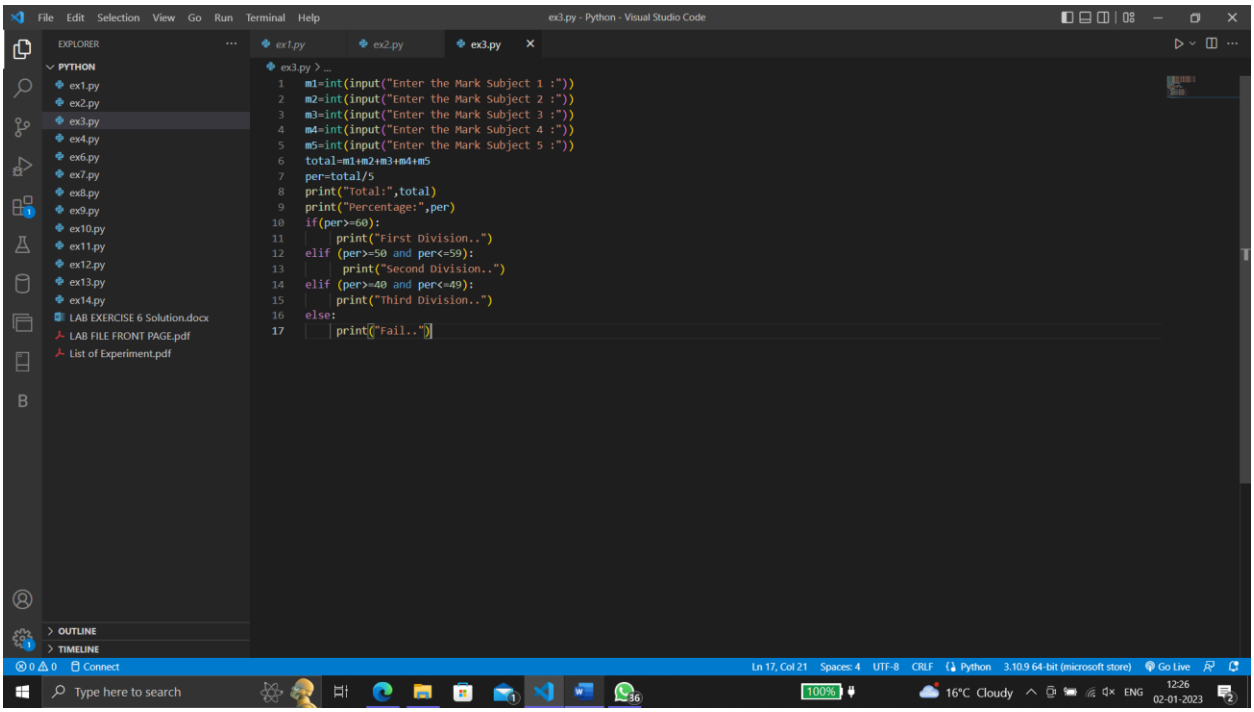
Aim: To write a Python program find the division of student.

Description:

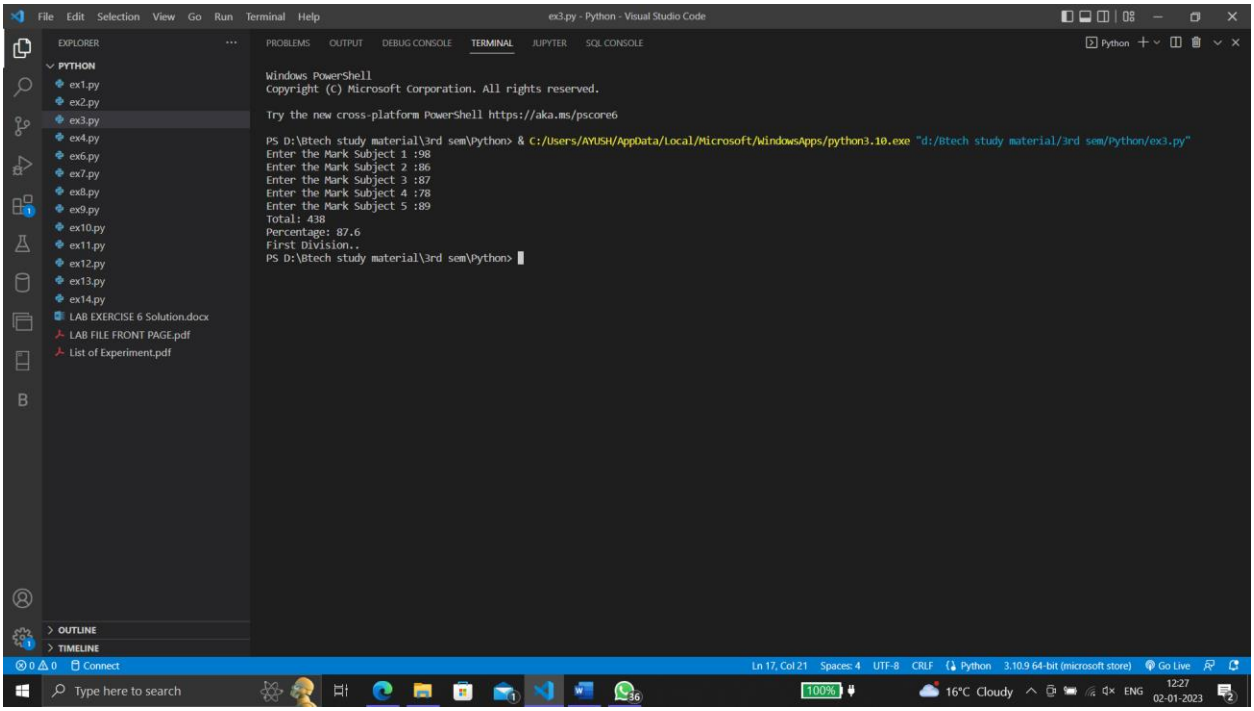
This program finds and prints the grade of a student based on marks obtained in five subjects entered by the user at run-time.
To calculate the grade of students in Python, you have to ask the user to enter marks obtained in five subjects. Now calculate the sum of all the marks and then calculate the average marks to find the grade according to the average marks obtained by the students.

Solution:

Code:



Input/Output:



LAB EXERCISE 6

Aim: To write a program implements Fibonacci series.

Description:

The Fibonacci numbers are the numbers in the following integer sequence. 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ……..

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

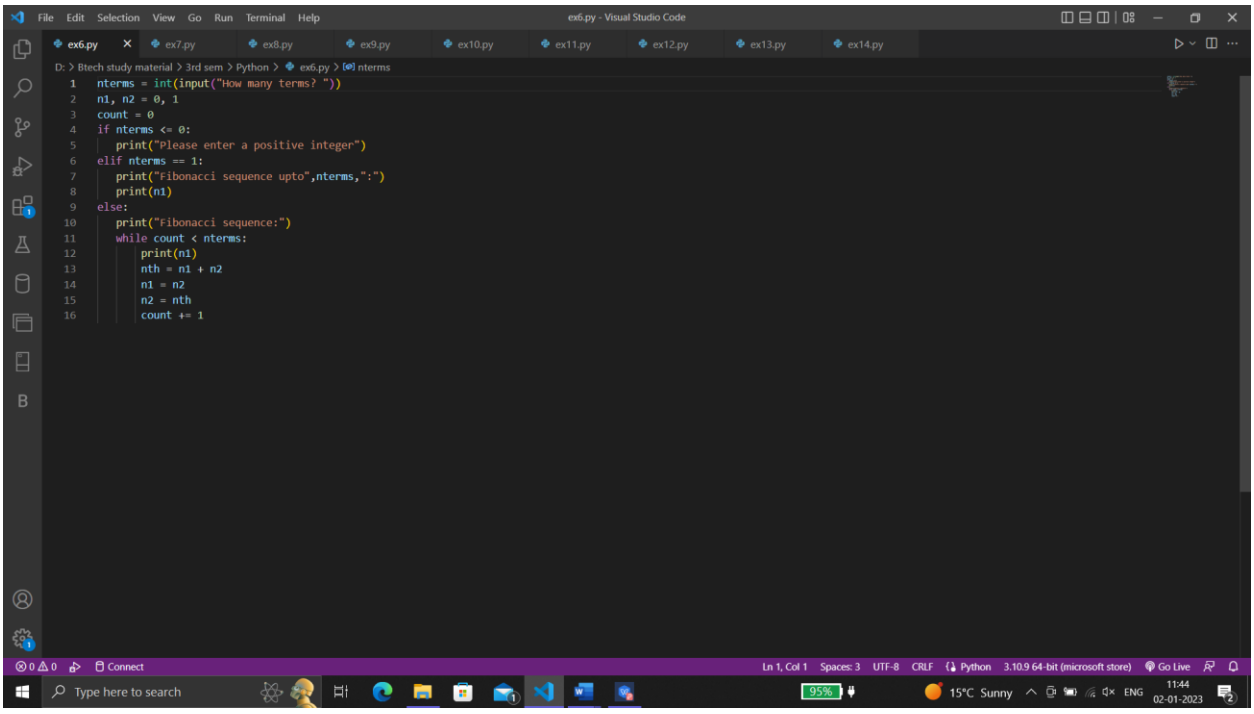
$F_n = F_{n-1} + F_{n-2}$

with seed values

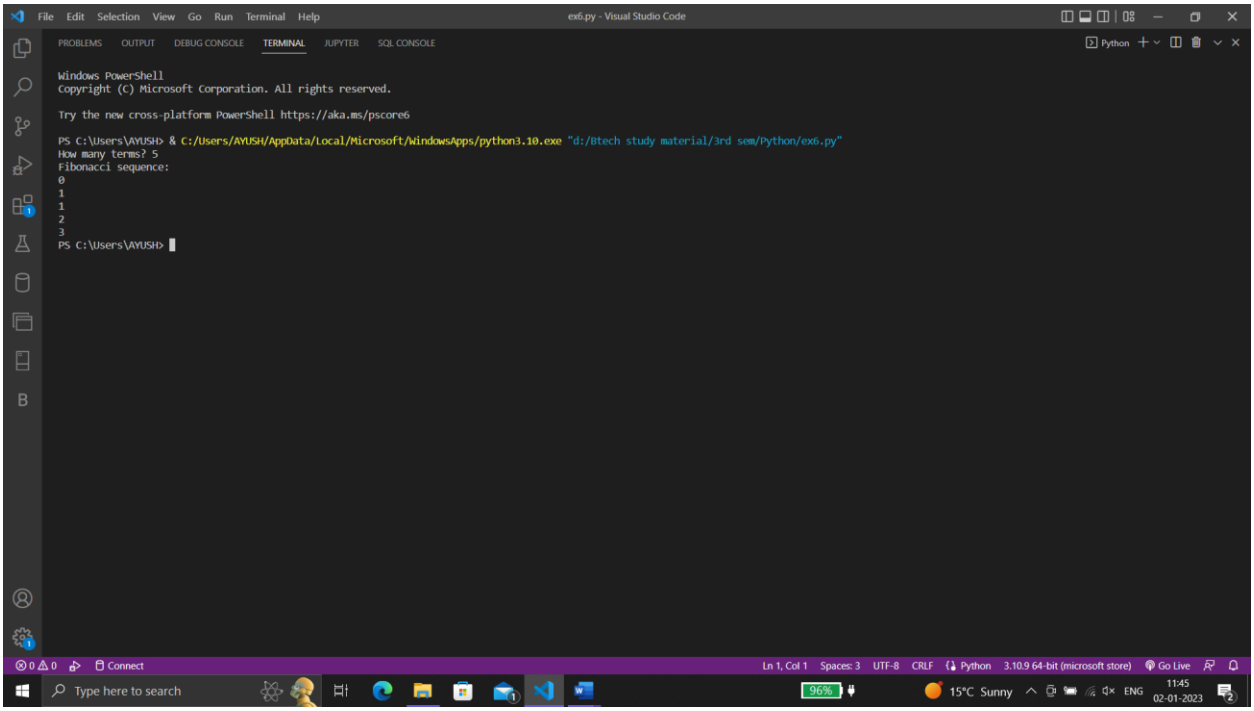
$F_0 = 0$ and $F_1 = 1$.

Solution:

Code:



Input/Output:



LAB EXERCISE 7

Aim: To write a Python program for factorial

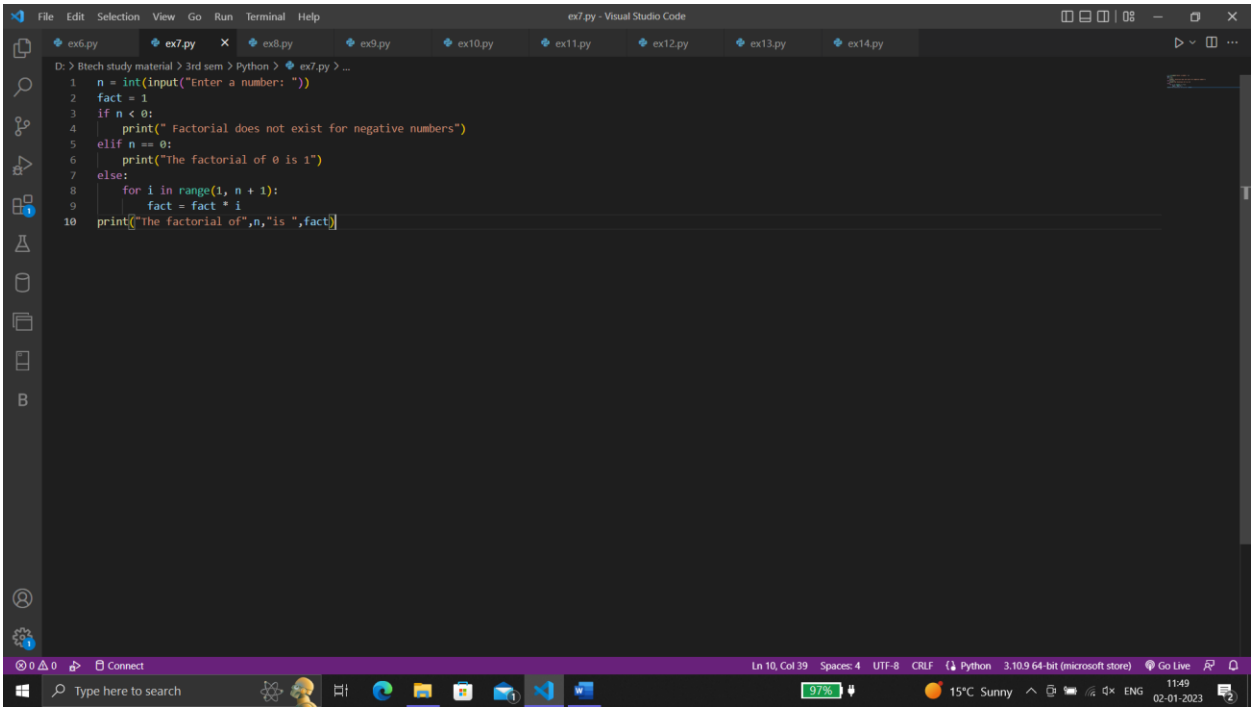
Description:

factorial is a simple thing. Factorials are just products. An exclamation mark indicates the factorial. Factorial is a multiplication operation of natural numbers with all the natural number that are less than it.

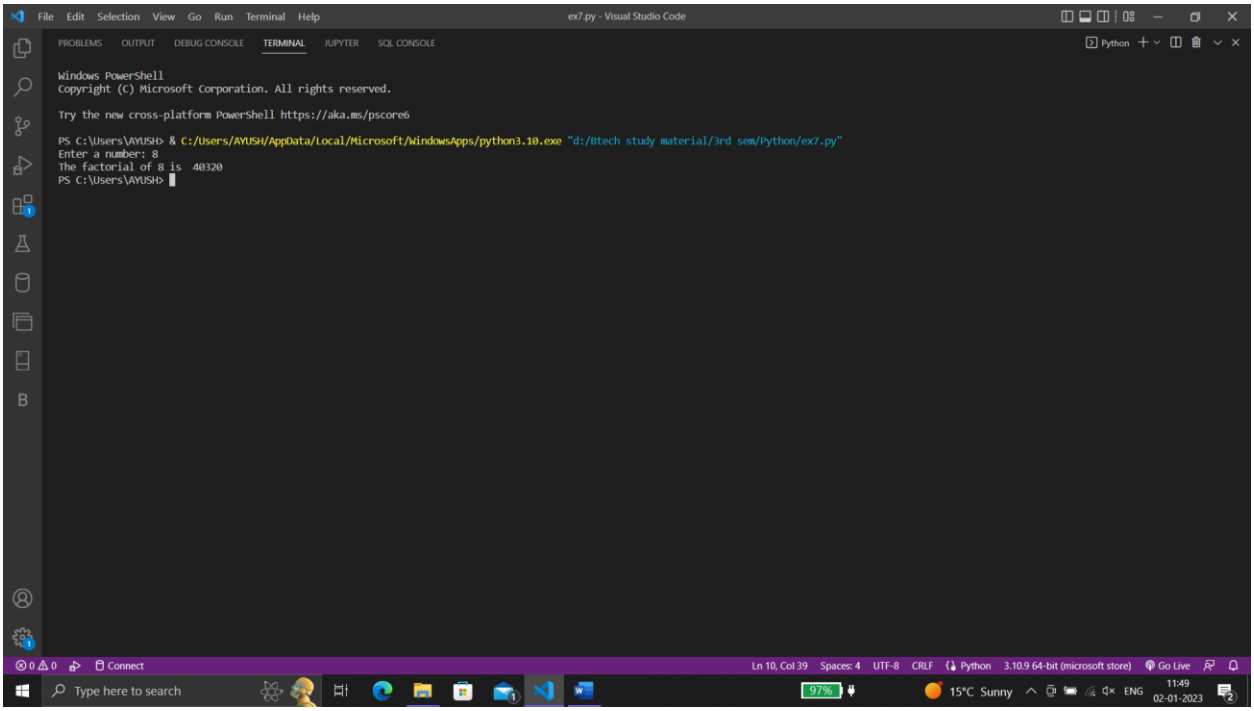
$$n! = n \times (n-1) \times (n-2) \times (n-3) \times \ldots \times 3 \times 2 \times 1$$

Solution:

Code:



Input/Output:



LAB EXERCISE 8

Aim: To write a Python program to use of functions

Description:

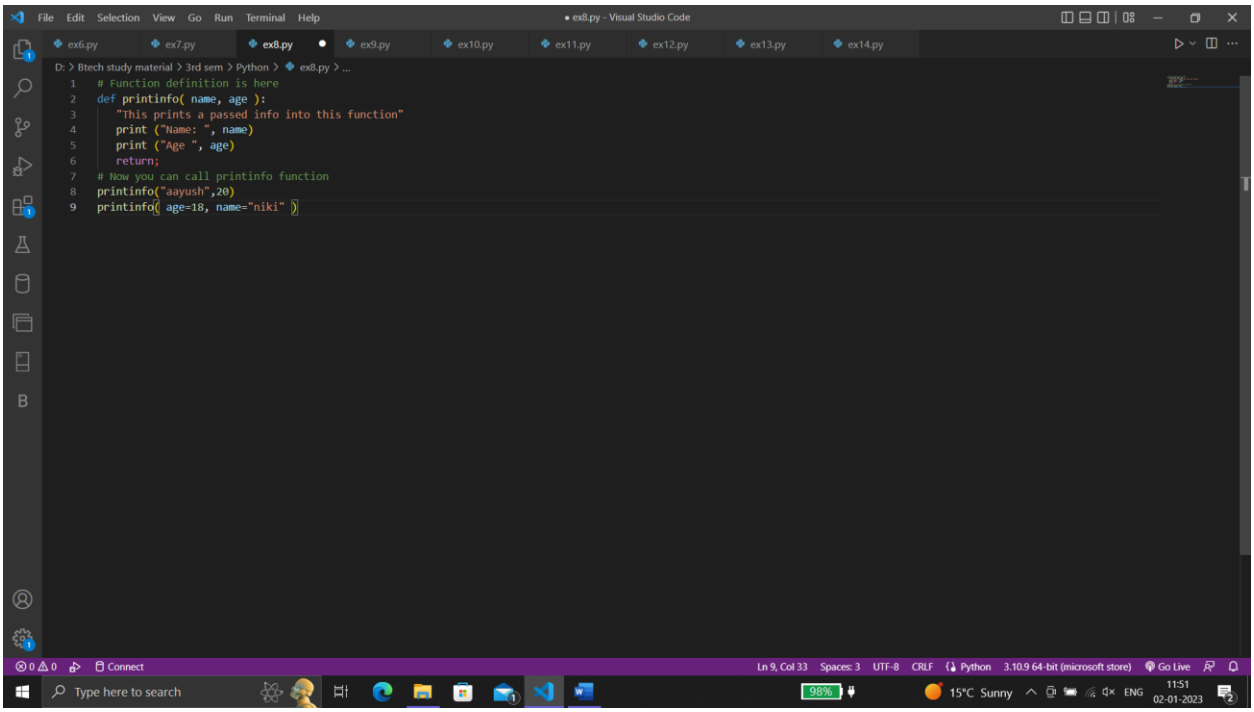
A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

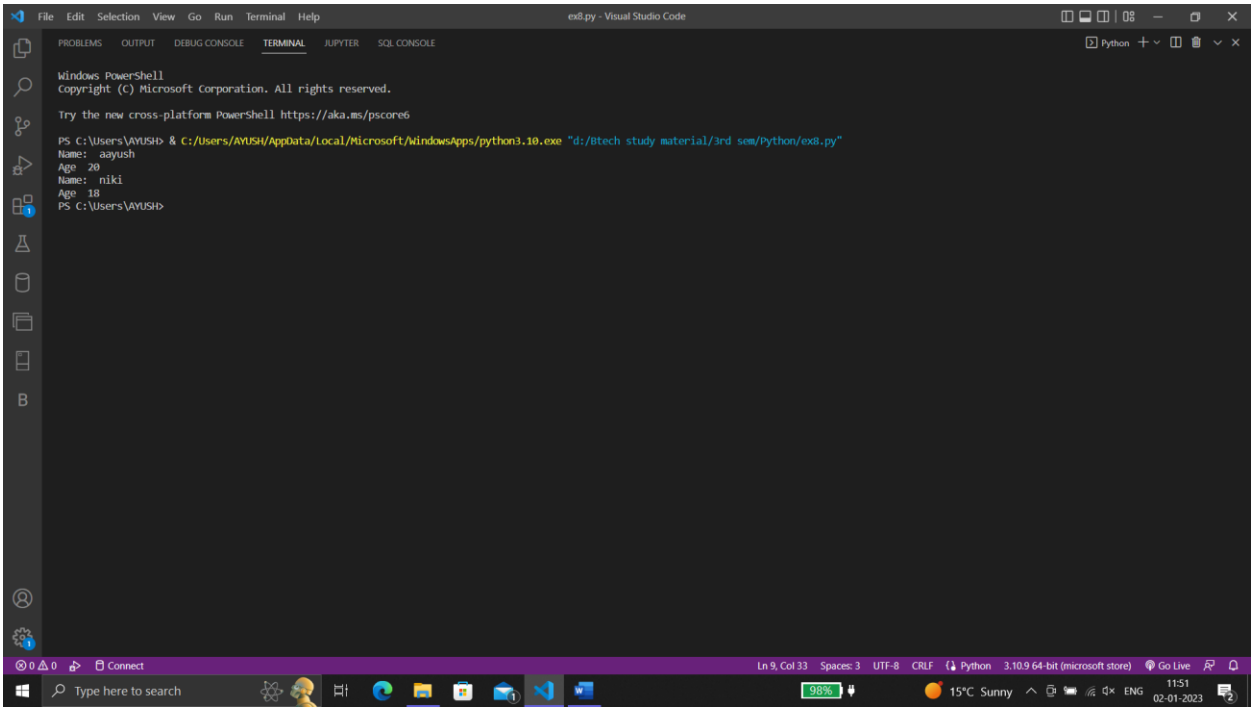
A function can return data as a result.

Solution:

Code:



Input/Output:



LAB EXERCISE 9

Aim: To write a Python program to implement list

Description:

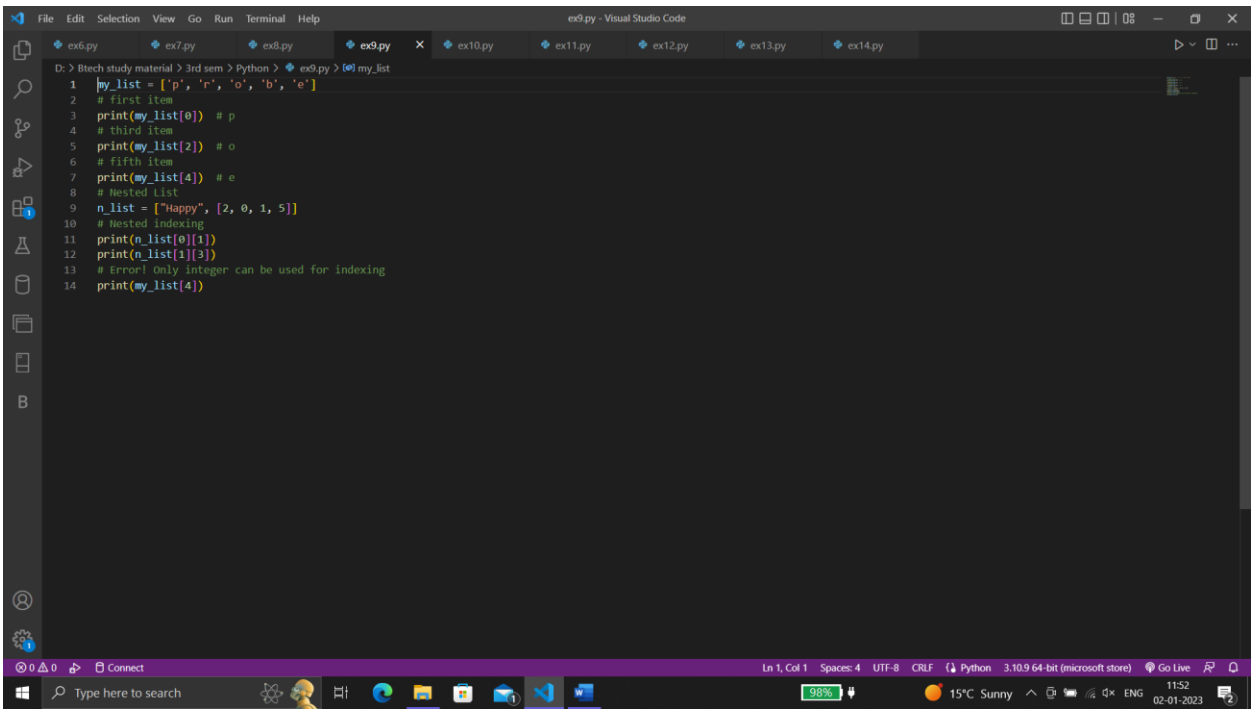
Lists are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

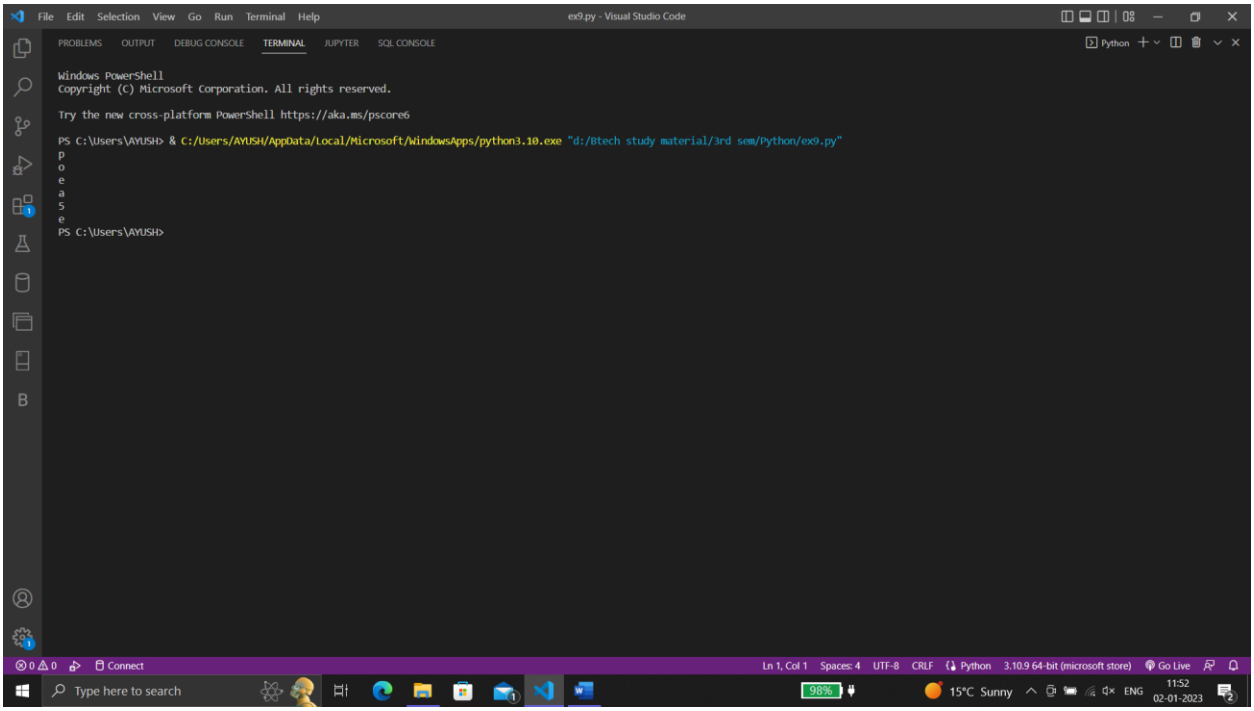
Lists are created using square brackets:

Solution:

Code:



Input/Output:



LAB EXERCISE 10

Aim: To write a Python program to implement tuples.

Description:

Tuples are used to store multiple items in a single variable.

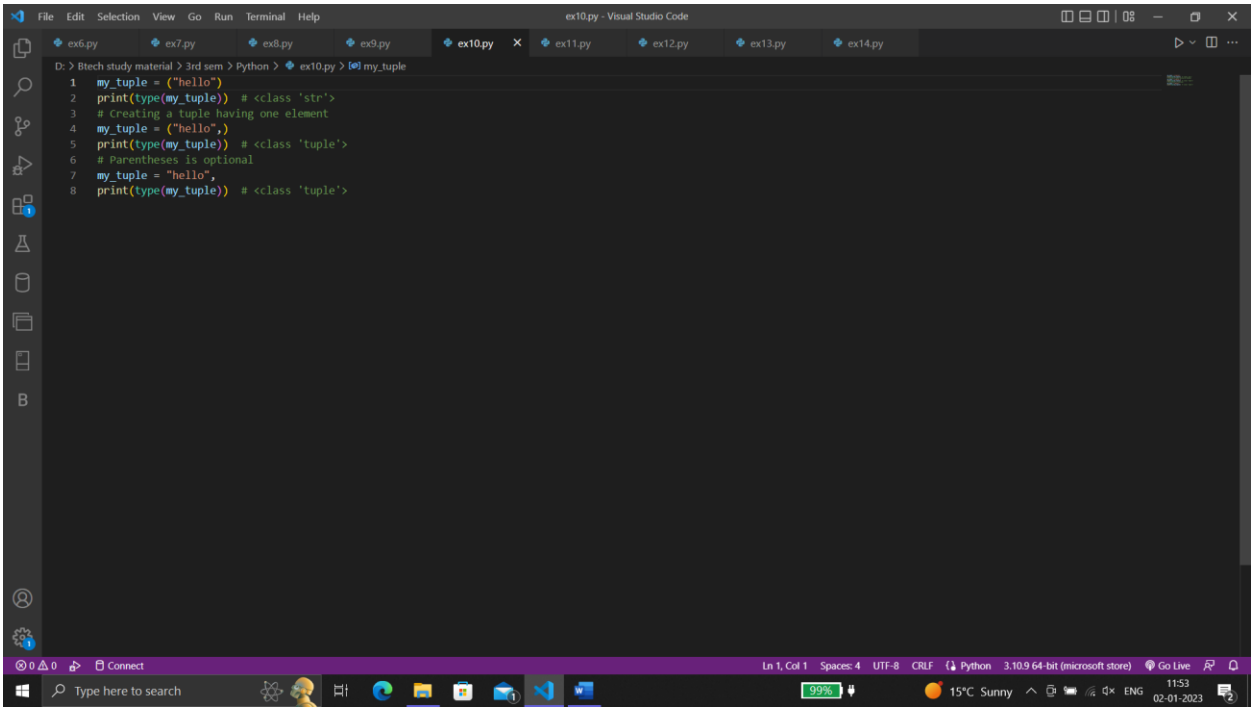
Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.

A tuple is a collection which is ordered and unchangeable**.**

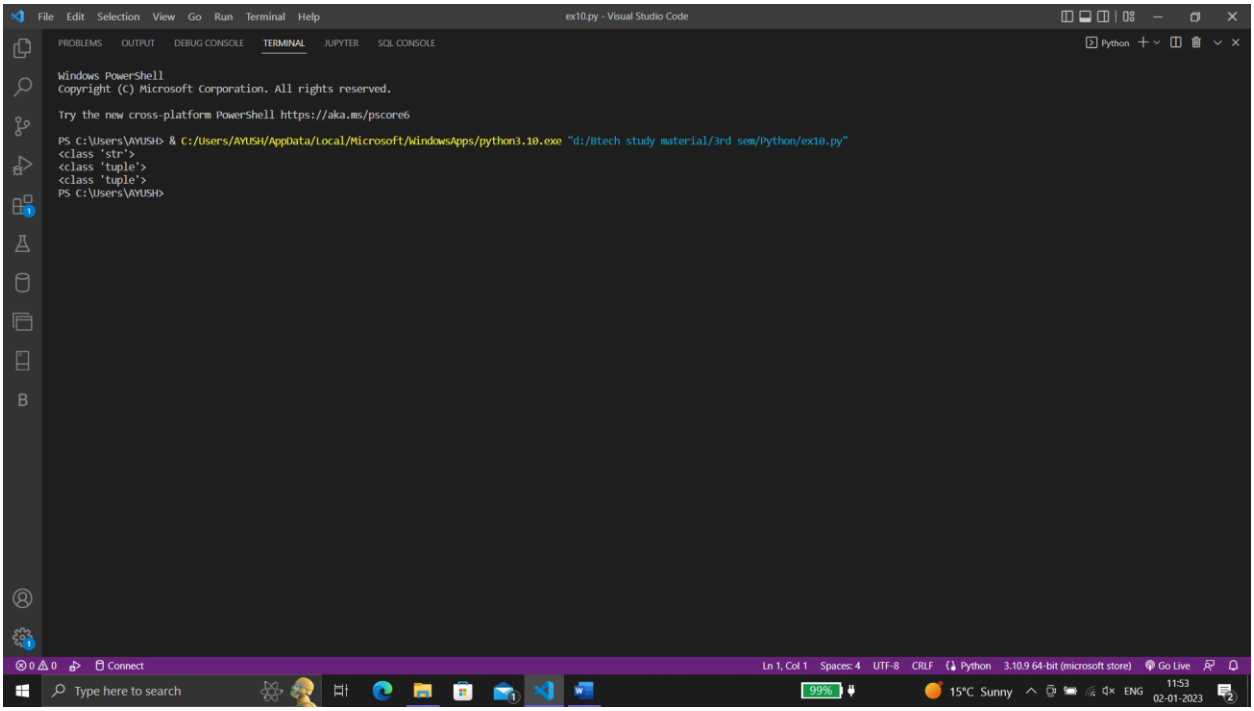
Tuples are written with round brackets.

Solution:

Code:



Input/Output:



LAB EXERCISE 11

Aim: To write a Python program Insertion sort.

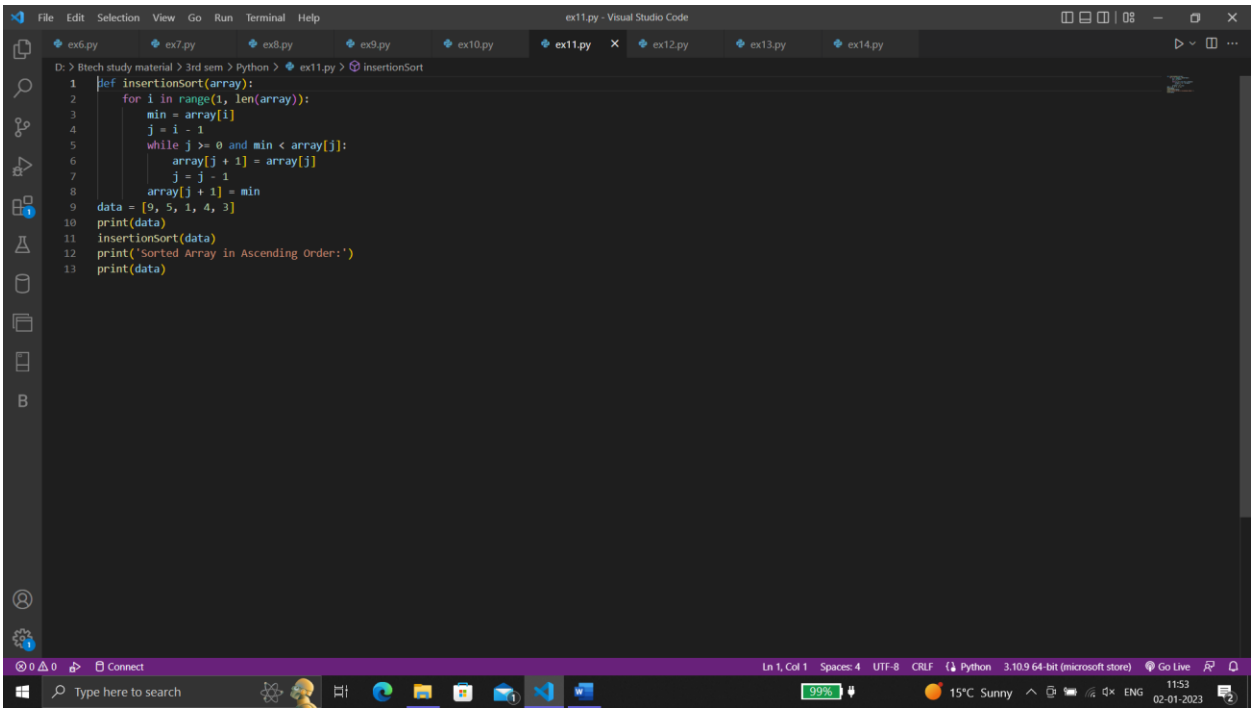
Description:

Insertion sort is the simple method of sorting an array. In this technique, the array is virtually split into the sorted and unsorted part. An element from unsorted part is picked and is placed at correct position in the sorted part.

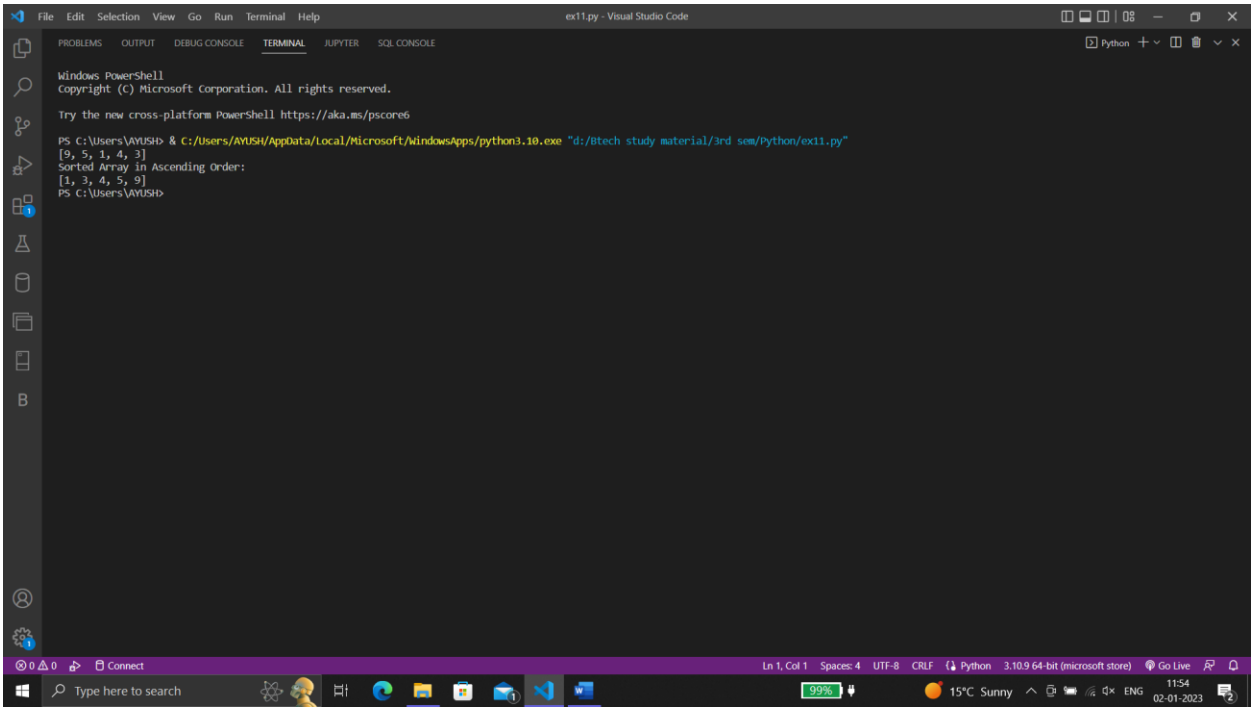
- The array elements are traversed from 1 to n.
- If the array element at position i is greater than its predecessor, it does not need to be moved.

Solution:

Code:



Input/Output:



LAB EXERCISE 12

Aim: To write a Python program merge sort.

Description:

Merge Sort is a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. The merge() function is used for merging two halves.

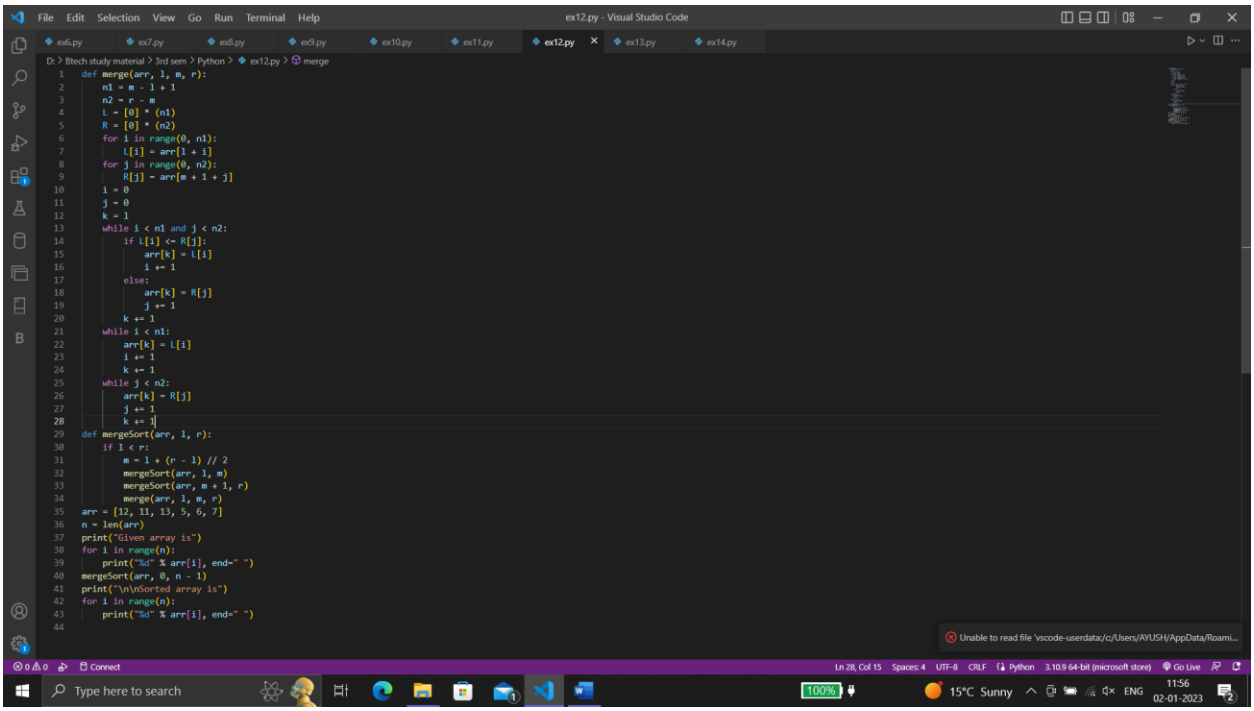
The merge(arr, l, m, r) is key process that assumes that arr[l..m] and arr[m+1..r] are sorted and merges the two sorted sub-arrays into one.

The sub lists are divided again and again into halves until we get the only one element each. Then we combine the pair of one element lists into two element lists, sorting them in the process.

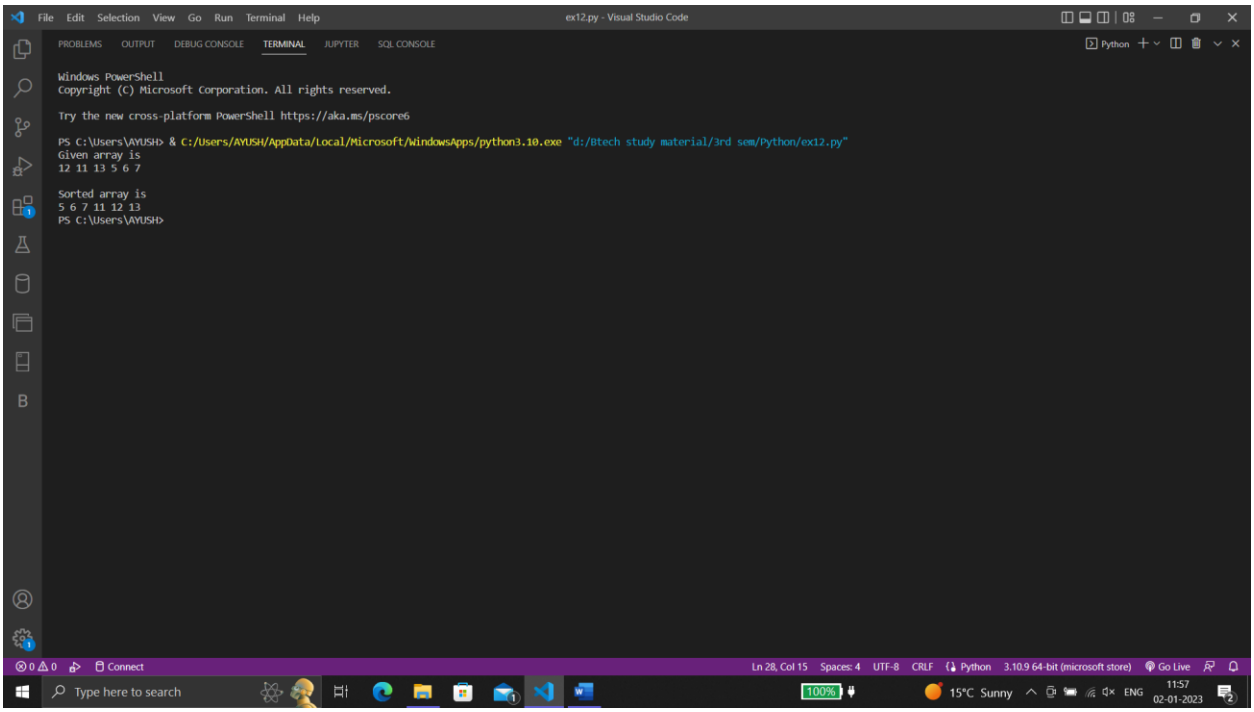
The sorted two element pairs is merged into the four element lists, and so on until we get the sorted list.

Solution:

Code:



Input/Output:



LAB EXERCISE 13

Aim: To write a Python program first n prime numbers

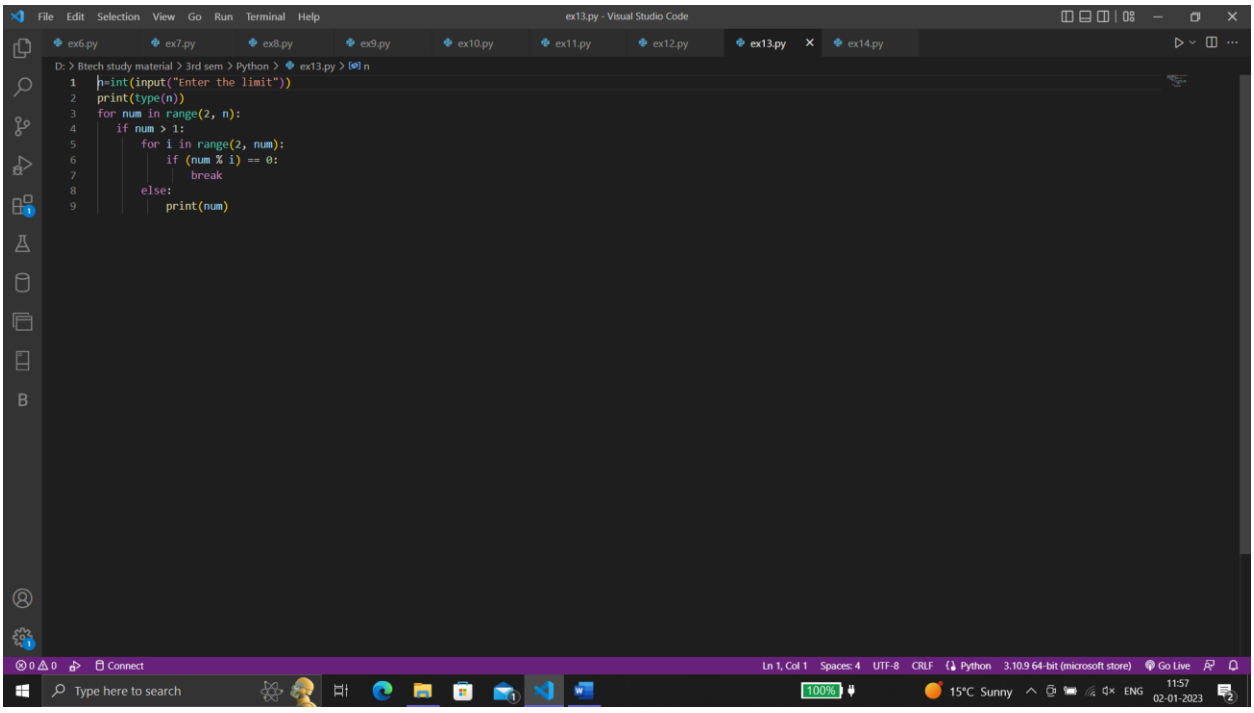
Description:

A prime number is a natural number which is greater than 1 and has no positive divisor other than 1 and itself, such as 2, 3, 5, 7, 11, 13, and so on.

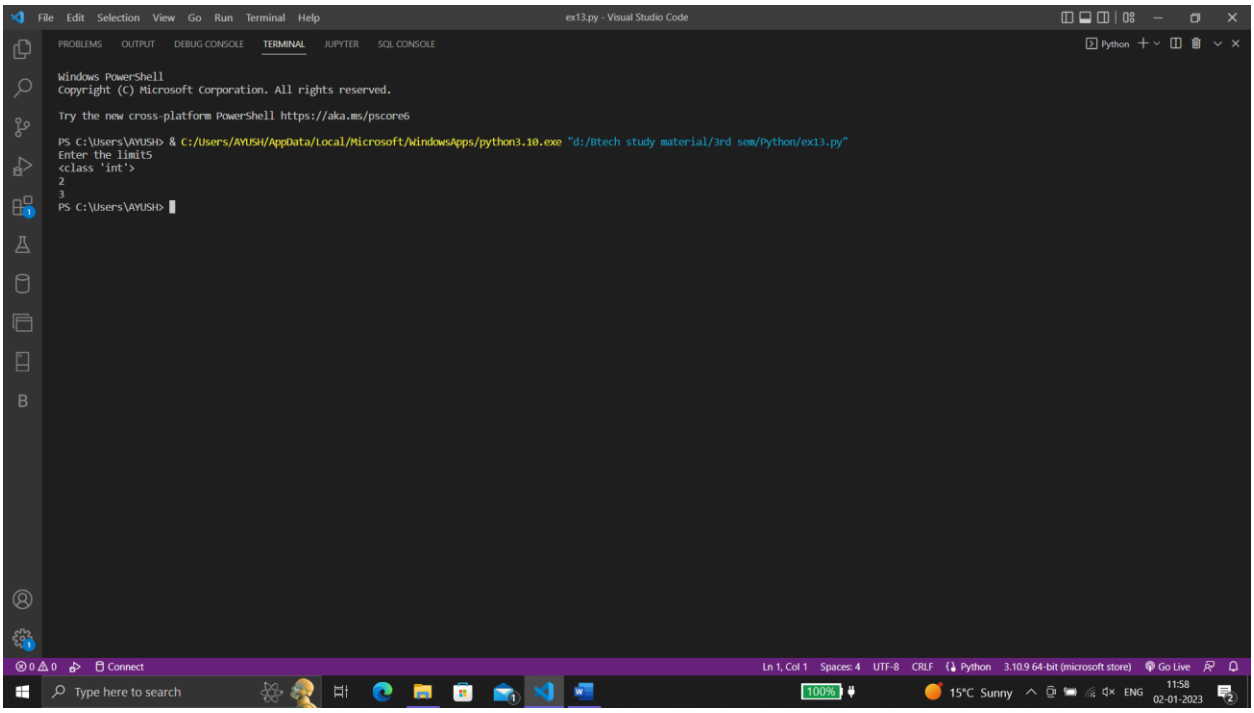
The user is given two integer numbers, lower value, and upper value. The task is to write the Python program for printing all the prime numbers between the given interval (or range).

Solution:

Code:



Input/Output:



LAB EXERCISE 14

Aim:-Implementation of Data Science concepts using Python

DESCRIPTION:-This Data Science with Python program provides learners with a complete understanding of data analytics tools & techniques. Getting started with Python can help you gain knowledge on data analysis, visualization, NumPy, SciPy, web scraping, and natural language processing.

Python offers a good number of libraries used in data science such as Pandas, Numpy, and Scikit-learn. Learning those libraries right away and skipping the basics isn't good though.

If you would like to learn Python for data science, you should master Python core concepts first. Having a solid foundation in Python will help you avoid common mistakes and bad practices. As a result, learning Python libraries used in data science will be much easier.

In this guide, we'll see some must-know Python concepts every data scientist should know. At the end of this article, you will find a Python for Data Science Cheat Sheet in PDF version (section 9 in the table of contents below)

1. Python Attributes vs Methods

I can't tell how long I used the words "attribute" and "method" interchangeably when I was a beginner in Python.

When you learn libraries like Pandas, you'll frequently call attributes and methods, so it's good to know what's the difference between them.

- Attribute: An attribute is a variable stored in a class. That is, a value associated with an object. It's referenced by name using dotted expressions. For an object `Foo`, you call an attribute `bar` as `Foo.bar`
- Method: A method is a function that is defined inside a class body. For an object `Foo`, you call a method `baz` as `Foo.baz()`