

## ASSIGNMENT-1

### 1.Check if Prime or Not

#### Program:

```
a = int(input("Enter the number to check if it is a prime : "))
```

```
if a > 1:
```

```
    for i in range(2, a):
```

```
        if (a % i) == 0:
```

```
            print(a, " is not a prime number")
```

```
            break
```

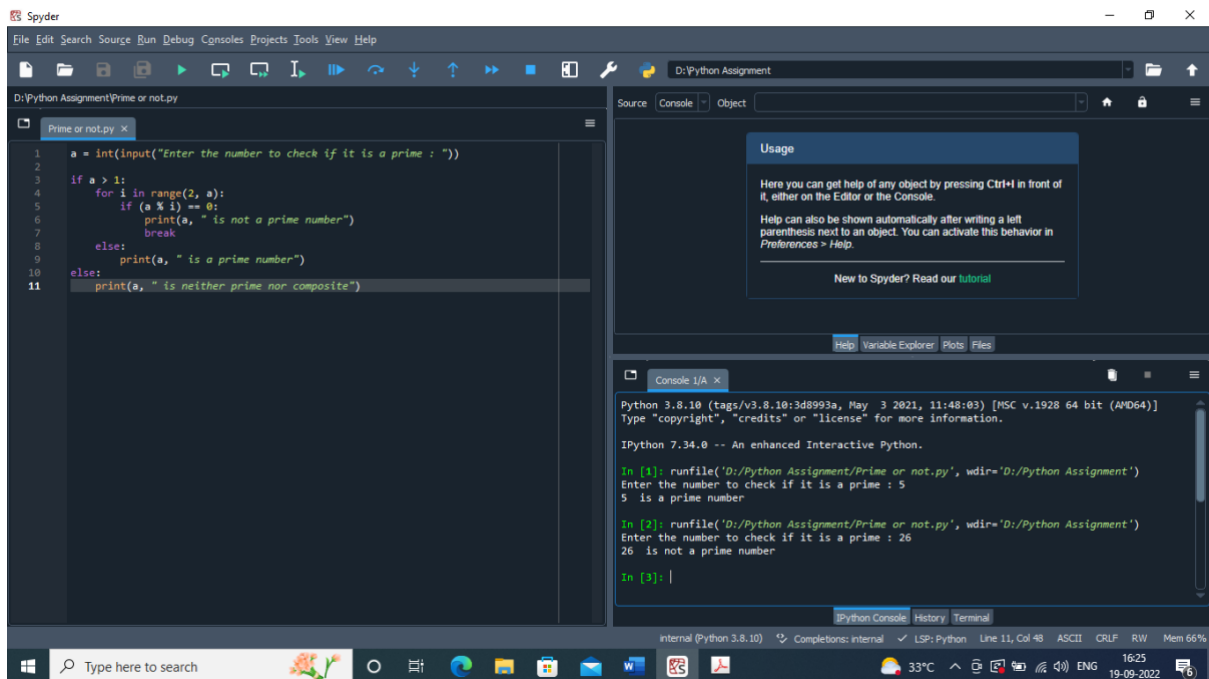
```
    else:
```

```
        print(a, " is a prime number")
```

```
else:
```

```
    print(a, " is neither prime nor composite")
```

#### Output:



The screenshot displays the Spyder Python IDE interface. The left pane shows the source code for a file named 'Prime or not.py'. The code is as follows:

```
1 a = int(input("Enter the number to check if it is a prime : "))
2
3 if a > 1:
4     for i in range(2, a):
5         if (a % i) == 0:
6             print(a, " is not a prime number")
7             break
8     else:
9         print(a, " is a prime number")
10
11 else:
12     print(a, " is neither prime nor composite")
```

The right pane shows the IPython console output. It includes the Python version (3.8.10), the Spyder version (7.34.0), and the execution results for three input values: 5, 26, and an empty input.

```
Python 3.8.10 (tags/v3.8.10:3d8992a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.34.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/Python Assignment/Prime or not.py', wdir='D:/Python Assignment')
Enter the number to check if it is a prime : 5
5 is a prime number

In [2]: runfile('D:/Python Assignment/Prime or not.py', wdir='D:/Python Assignment')
Enter the number to check if it is a prime : 26
26 is not a prime number

In [3]:
```

The bottom status bar indicates the current file is 'internal (Python 3.8.10)' and shows the cursor position at Line 11, Col 48.

## 2.Generate odd number from m to n using while loop:

### Program:

```
print("Finding odd numbers in a given range....")
```

```
m = int(input("From : "))
```

```
n = int(input("To :"))
```

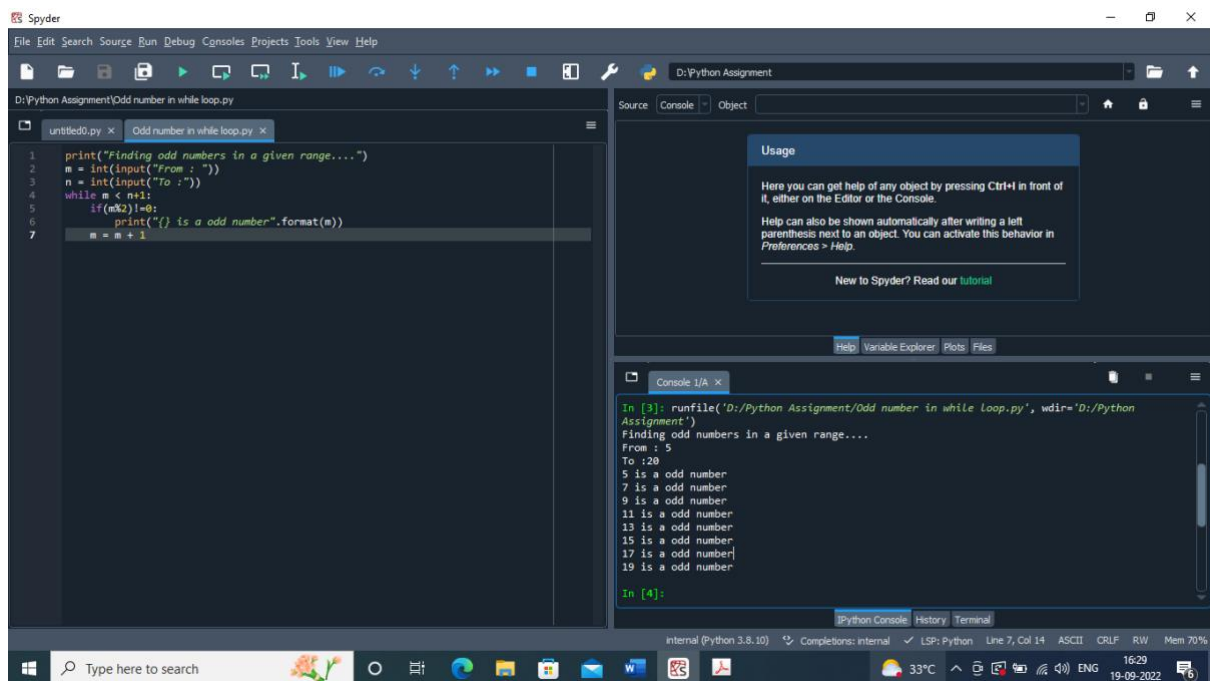
```
while m < n+1:
```

```
    if(m%2)!=0:
```

```
        print("{} is a odd number".format(m))
```

```
    m = m + 1
```

### Output:



The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script named 'Odd number in while loop.py' with the following code:

```
1 print("Finding odd numbers in a given range....")
2 m = int(input("From : "))
3 n = int(input("To :"))
4 while m < n+1:
5     if(m%2)!=0:
6         print("{} is a odd number".format(m))
7     m = m + 1
```

The right-hand pane contains a 'Console' window showing the output of the program. The output is as follows:

```
In [3]: runfile('D:/Python Assignment/Odd number in while Loop.py', wdir='D:/Python
Assignment')
Finding odd numbers in a given range....
From : 5
To : 20
5 is a odd number
7 is a odd number
9 is a odd number
11 is a odd number
13 is a odd number
15 is a odd number
17 is a odd number
19 is a odd number
In [4]:
```

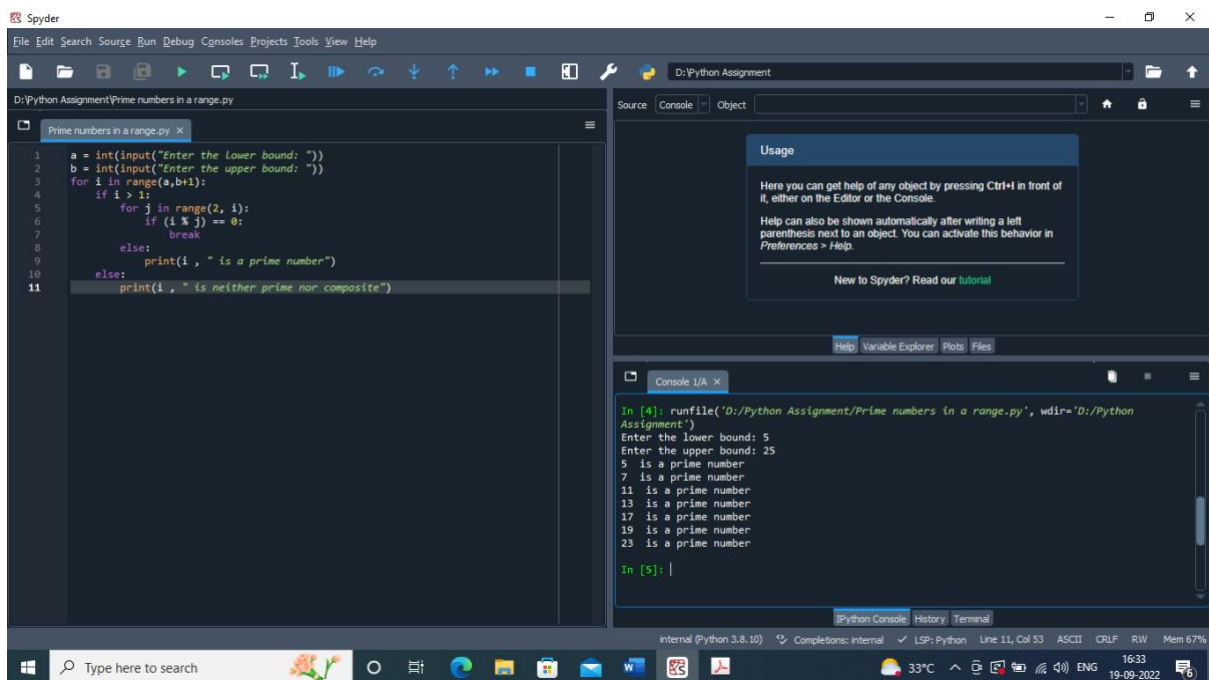
The bottom status bar indicates the Python version is 3.8.10 and the file is at Line 7, Col 14.

### 3.Display Prime Number Series Upto Given Number:

#### Program:

```
a = int(input("Enter the lower bound: "))
b = int(input("Enter the upper bound: "))
for i in range(a,b+1):
    if i > 1:
        for j in range(2, i):
            if (i % j) == 0:
                break
        else:
            print(i , " is a prime number")
    else:
        print(i , " is neither prime nor composite")
```

#### Output:



## 4.Generate Fibonacci Series:

### Program:

```
a = 0
```

```
b = 1
```

```
n = int(input("Enter the range of fibonacci numbers you wish to find : "))
```

```
print(a)
```

```
print(b)
```

```
for i in range(0,n-2):
```

```
    fib = a + b
```

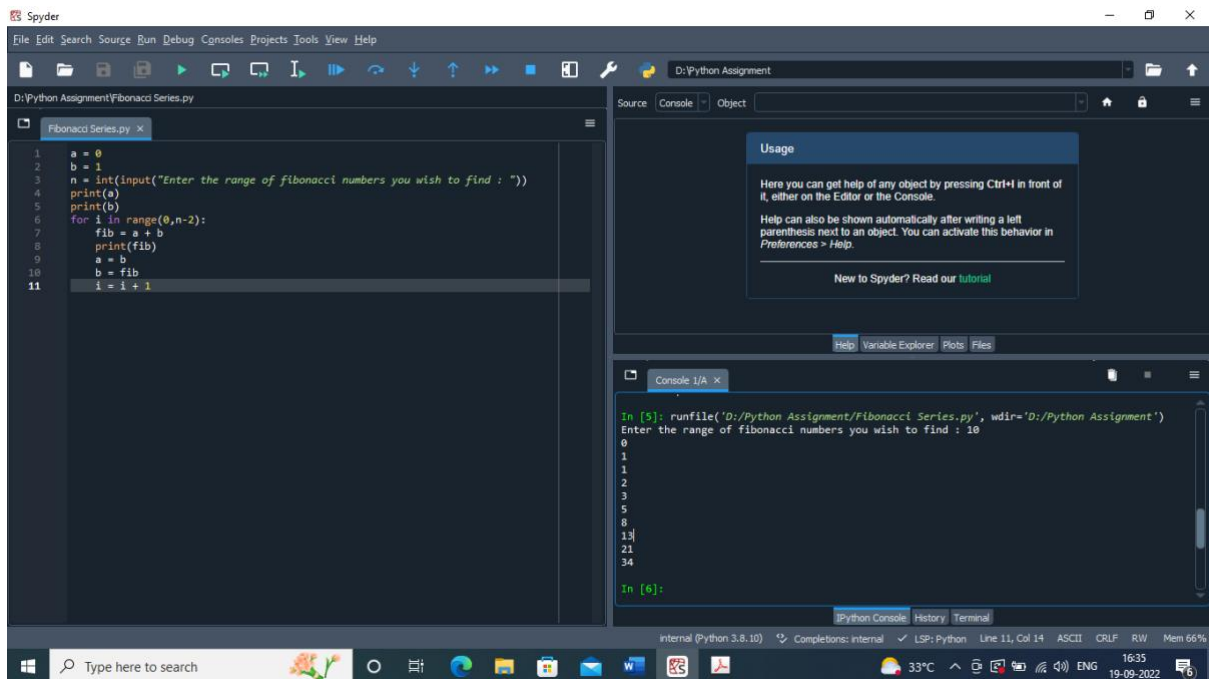
```
    print(fib)
```

```
    a = b
```

```
    b = fib
```

```
    i = i + 1
```

### Output:



The screenshot displays the Spyder Python IDE interface. The left pane shows the code for 'Fibonacci Series.py' with line numbers 1 through 11. The code initializes 'a' to 0 and 'b' to 1, prompts the user for a range 'n', and prints the first two numbers. A loop then generates and prints the subsequent Fibonacci numbers up to 'n-2', updating 'a' and 'b' in the process. The right pane is divided into two sections: the top section shows a 'Usage' help message, and the bottom section shows the 'Console' output. The console output displays the prompt 'Enter the range of fibonacci numbers you wish to find : 10' followed by the sequence of Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, and 34. The bottom status bar indicates the Python version is 3.8.10 and the file encoding is UTF-8.

```
1 a = 0
2 b = 1
3 n = int(input("Enter the range of fibonacci numbers you wish to find : "))
4 print(a)
5 print(b)
6 for i in range(0,n-2):
7     fib = a + b
8     print(fib)
9     a = b
10    b = fib
11    i = i + 1
```

Console 1/A x

```
In [5]: runfile('D:/Python Assignment/Fibonacci Series.py', wdir='D:/Python Assignment')
Enter the range of fibonacci numbers you wish to find : 10
0
1
1
2
3
5
8
13
21
34
In [6]:
```