

# 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
```

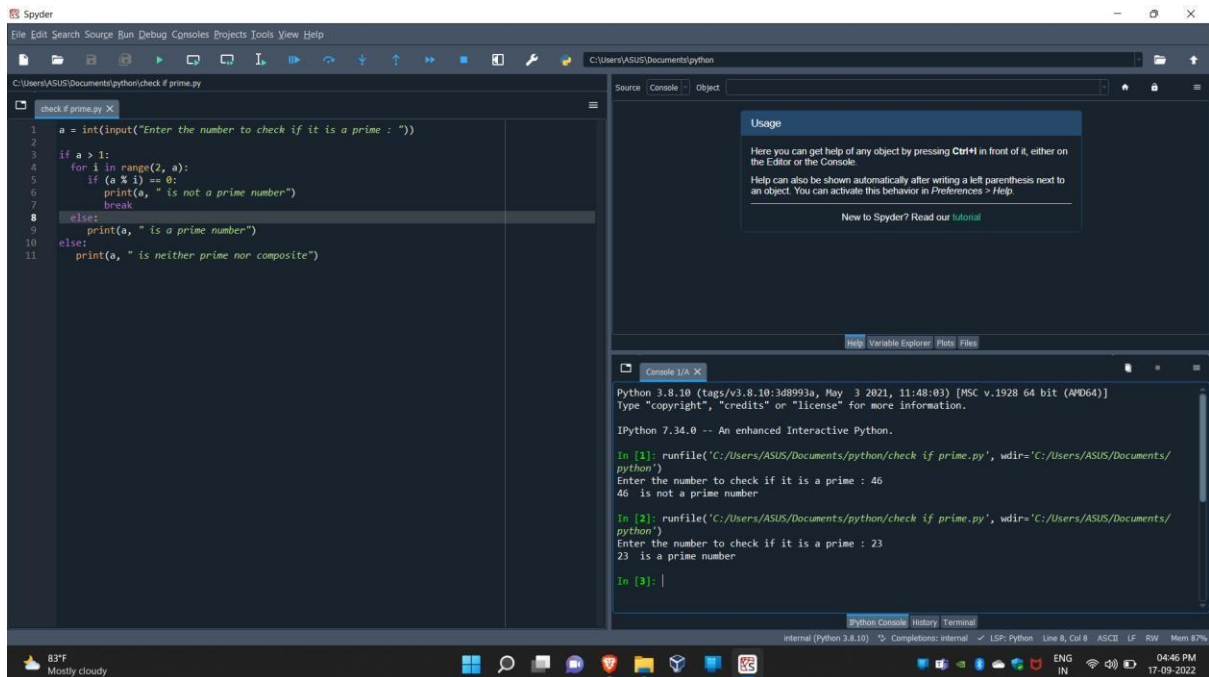
```
else:
```

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

```
else:
```

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

### Output:



The screenshot shows the Spyder Python IDE interface. The left pane displays the code for checking if a number is prime. The right pane shows the console output, which includes the program's execution and the user's input for two different numbers (46 and 23).

```
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")
```

Console Output:

```
Python 3.8.10 (tags/v3.8.10:3d8993a, 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('C:/Users/ASUS/Documents/python/check if prime.py', wdir='C:/Users/ASUS/Documents/python')
Enter the number to check if it is a prime : 46
46 is not a prime number

In [2]: runfile('C:/Users/ASUS/Documents/python/check if prime.py', wdir='C:/Users/ASUS/Documents/python')
Enter the number to check if it is a prime : 23
23 is a prime number

In [3]:
```

## 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 :"))

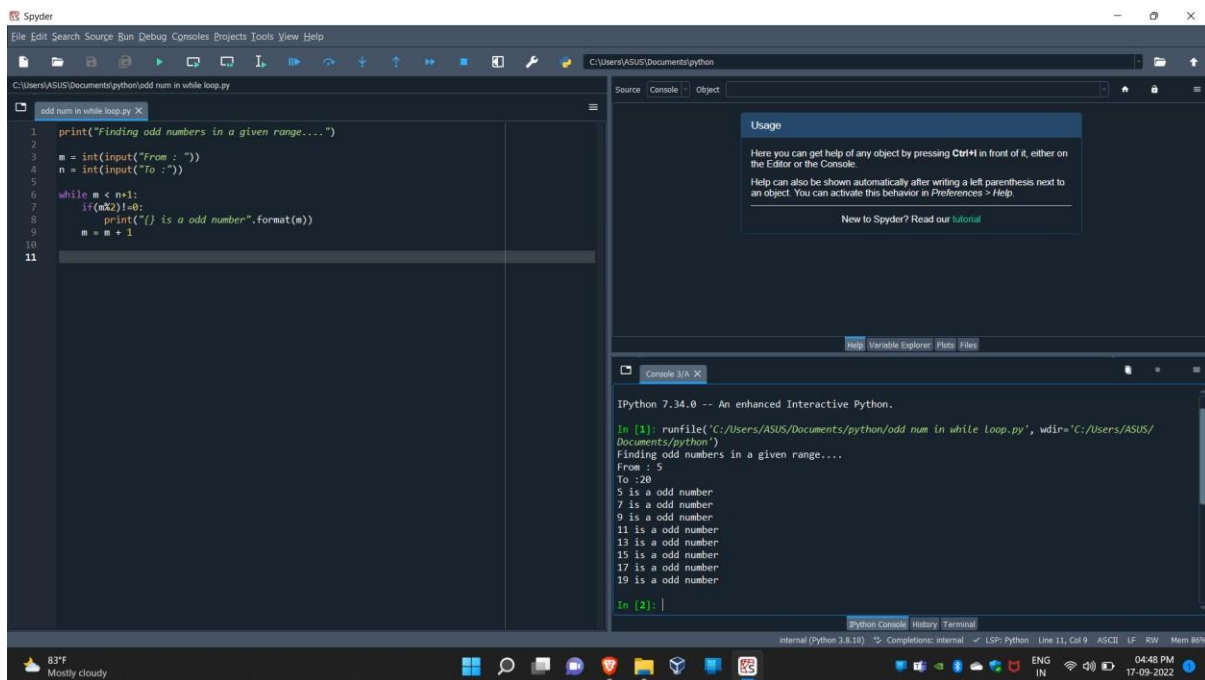
while m < n+1:

    if(m%2)!=0:

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

    m = m + 1
```

### Output:



The screenshot shows the Spyder Python IDE interface. The editor on the left contains the following Python code:

```
1 print("Finding odd numbers in a given range....")
2
3 m = int(input("From : "))
4 n = int(input("To :"))
5
6 while m < n+1:
7     if(m%2)!=0:
8         print("{} is a odd number".format(m))
9     m = m + 1
10
11
```

The right-hand pane shows the IPython console output:

```
IPython 7.34.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/ASUS/Documents/python/odd num in while loop.py', wdir='C:/Users/ASUS/
Documents/python')
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 [2]:
```

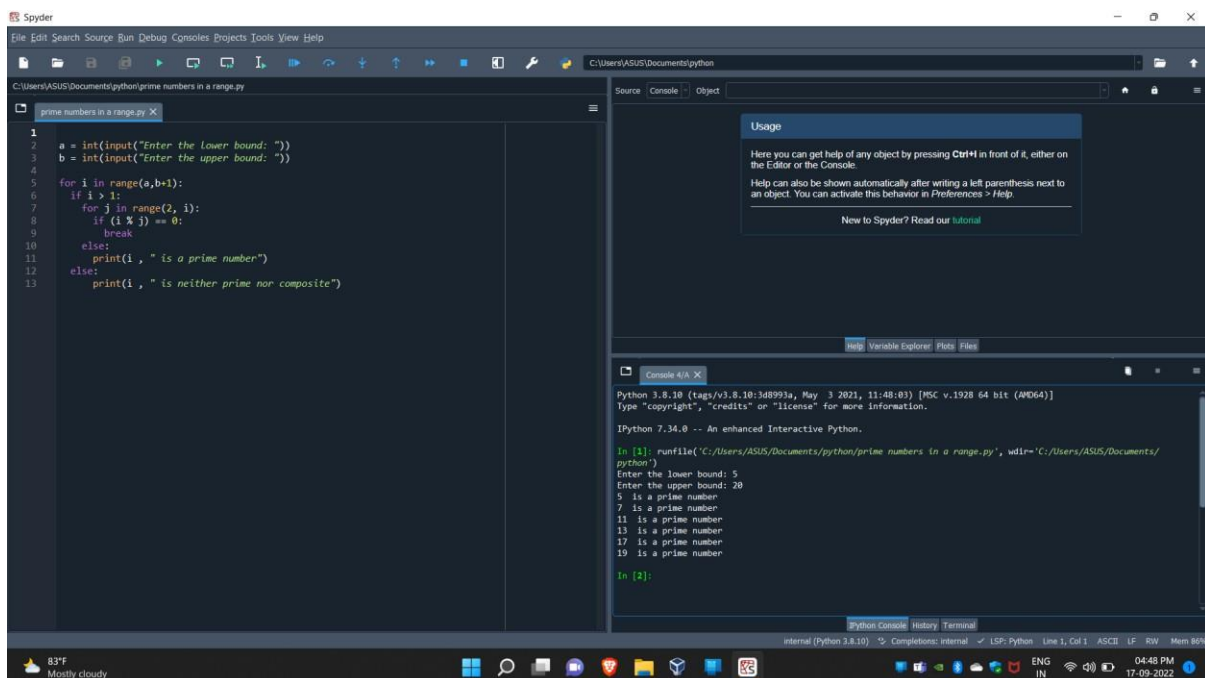
The bottom status bar indicates the environment is 'Internal (Python 3.8.10)' and the file is 'LSP: Python'.

### 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:



The screenshot displays the Spyder Python IDE interface. The main editor window on the left contains a Python script named 'prime numbers in a range.py'. The script prompts the user for a lower bound and an upper bound, then iterates through the range to identify prime numbers. The console window on the right shows the execution output, where the user has entered a lower bound of 5 and an upper bound of 20. The program correctly identifies and prints the prime numbers in this range: 5, 7, 11, 13, 17, and 19.

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\ASUS\Documents\python\prime numbers in a range.py
1 a = int(input("Enter the lower bound: "))
2 b = int(input("Enter the upper bound: "))
3
4
5 for i in range(a,b+1):
6     if i > 1:
7         for j in range(2, i):
8             if (i % j) == 0:
9                 break
10        else:
11            print(i , " is a prime number")
12    else:
13        print(i , " is neither prime nor composite")

Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+H in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial

Help Variable Explorer Plots File
Console 4/4 X
Python 3.8.10 (tags/v3.8.10:3d8993a, 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('C:/Users/ASUS/Documents/python/prime numbers in a range.py', wdir='C:/Users/ASUS/Documents/python')
Enter the lower bound: 5
Enter the upper bound: 20
5 is a prime number
7 is a prime number
11 is a prime number
13 is a prime number
17 is a prime number
19 is a prime number
In [2]:
```

83°F Mostly cloudy 04:48 PM 17-09-2022

## 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
```

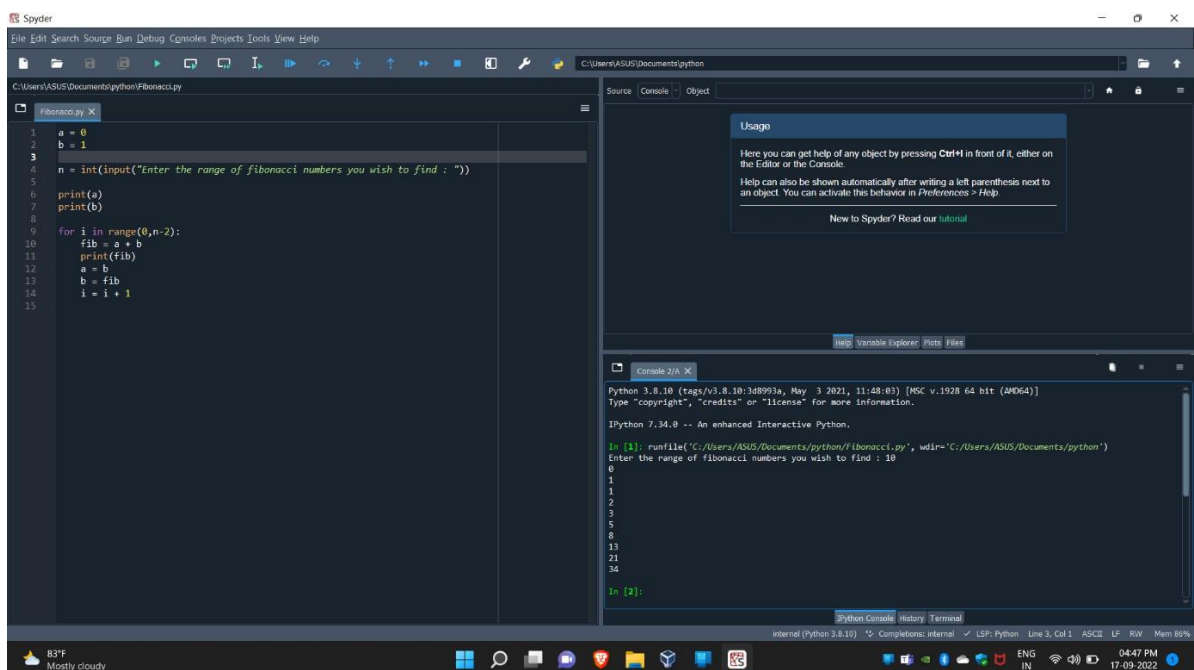
```
    print(fib)
```

```
    a = b
```

```
    b = fib
```

```
    i = i + 1
```

### Output:



The screenshot shows the Spyder Python IDE interface. The left pane displays the code for generating the Fibonacci series. The right pane is split into two sections: the top section shows a 'Usage' message, and the bottom section shows the console output. The console output displays the sequence of Fibonacci numbers generated for the input range of 10.

```
1 a = 0
2 b = 1
3
4 n = int(input("Enter the range of fibonacci numbers you wish to find : "))
5
6 print(a)
7 print(b)
8
9 for i in range(0,n-2):
10     fib = a + b
11     print(fib)
12     a = b
13     b = fib
14     i = i + 1
15
```

Usage

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

[New to Spyder? Read our tutorial](#)

Python 3.8.10 (tags/v3.8.10:3d89923a, May 3 2021, 11:45:03) [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('C:/Users/ASUS/Documents/python/fibonacci.py', wdir='C:/Users/ASUS/Documents/python')
Enter the range of fibonacci numbers you wish to find : 10
0
1
1
2
3
5
8
13
21
34

In [2]:
```

83°F  
Mostly cloudy

04:47 PM  
17-09-2022