

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 Python code for checking if a number is prime. The right pane shows the output of the program, including the definition of the integer type and the execution results for two test cases: 46 (not prime) and 29 (prime).

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

Definition: `j [x]` -> Integer: `int x, base 10` -> Integer

Convert a number or string to an integer, or return 0 if no arguments are given. If `x` is a number, return `x.__int__()`. For floating point numbers, this truncates towards zero.

If `x` is not a number or if `base` is given, then `x` must be a string, bytes, or bytearray instance representing an integer literal in the given base. The literal can be preceded by '+' or '-' and be surrounded by whitespace. The base defaults to 10. Valid bases are 0 and 2-36. Base 0 means to interpret the base from the string as an integer literal. >>> `int('0b100', base=0)` 4

Python 3.8.10 (tags/v3.8.10:3d8991a, May 3 2021, 11:48:01) [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 : 29
29 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 :"))

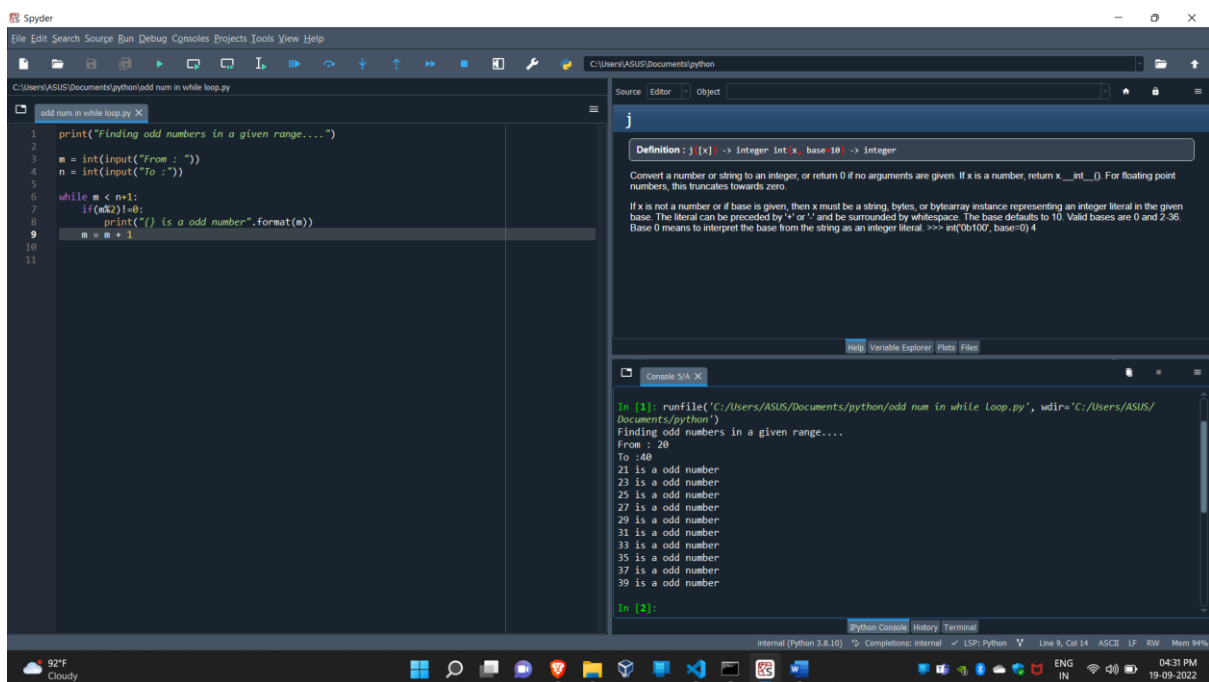
while m < n+1:

    if(m%2)!=0:

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

    m = m + 1
```

Output:



The screenshot displays the Spyder Python IDE interface. The left pane shows the source code for a file named 'odd num in while loop.py'. The code prompts the user for a range and prints odd numbers within that range. The right pane shows the Python documentation for the 'int()' function. The bottom pane shows the console output, which includes the program's execution and the list of odd numbers from 21 to 39.

```
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
```

Definition: `int(x) -> Integer` `int(x, base=10) -> Integer`

Convert a number or string to an integer, or return 0 if no arguments are given. If x is a number, return `x.__int__()`. For floating point numbers, this truncates towards zero.

If x is not a number or if base is given, then x must be a string, bytes, or bytearray instance representing an integer literal in the given base. The literal can be preceded by '+' or '-' and be surrounded by whitespace. The base defaults to 10. Valid bases are 0 and 2-36. Base 0 means to interpret the base from the string as an integer literal. >>> `int('0b100', base=0)` 4

```
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 : 20
To : 40
21 is a odd number
23 is a odd number
25 is a odd number
27 is a odd number
29 is a odd number
31 is a odd number
33 is a odd number
35 is a odd number
37 is a odd number
39 is a odd number

In [2]:
```

internal (Python 3.8.10) Completions: internal LSP: Python Line 9, Col 14 ASCE LF RW Mem 94%

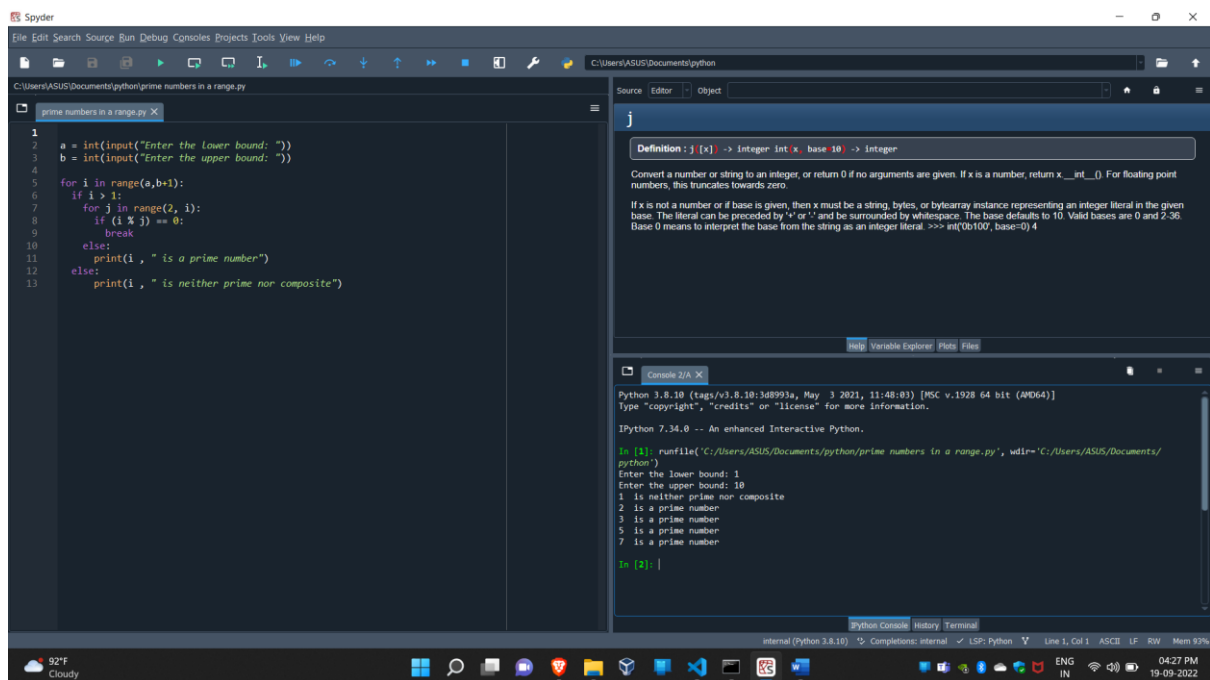
92°F Cloudy 04:31 PM 19-09-2022

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 left pane shows the source code for a program that prints prime numbers between two user-defined bounds. The right pane is divided into two sections: the top section shows the definition of the variable `j` as an integer, and the bottom section shows the Python console output. The console output indicates that the program was run successfully, and it displays the prime numbers 2, 3, 5, and 7 for the input range 1 to 10.

```
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")
```

Definition: `j [x] -> Integer` `int x, base 10 -> Integer`

Convert a number or string to an integer, or return 0 if no arguments are given. If x is a number, return `x.__int__()`. For floating point numbers, this truncates towards zero.

If x is not a number or if base is given, then x must be a string, bytes, or bytearray instance representing an integer literal in the given base. The literal can be preceded by '+' or '-' and be surrounded by whitespace. The base defaults to 10. Valid bases are 0 and 2-36. Base 0 means to interpret the base from the string as an integer literal. `>>> int('0b100', base=0) 4`

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.

Python 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: 1
Enter the upper bound: 10
1 is neither prime nor composite
2 is a prime number
3 is a prime number
5 is a prime number
7 is a prime number

In [2]:
```

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

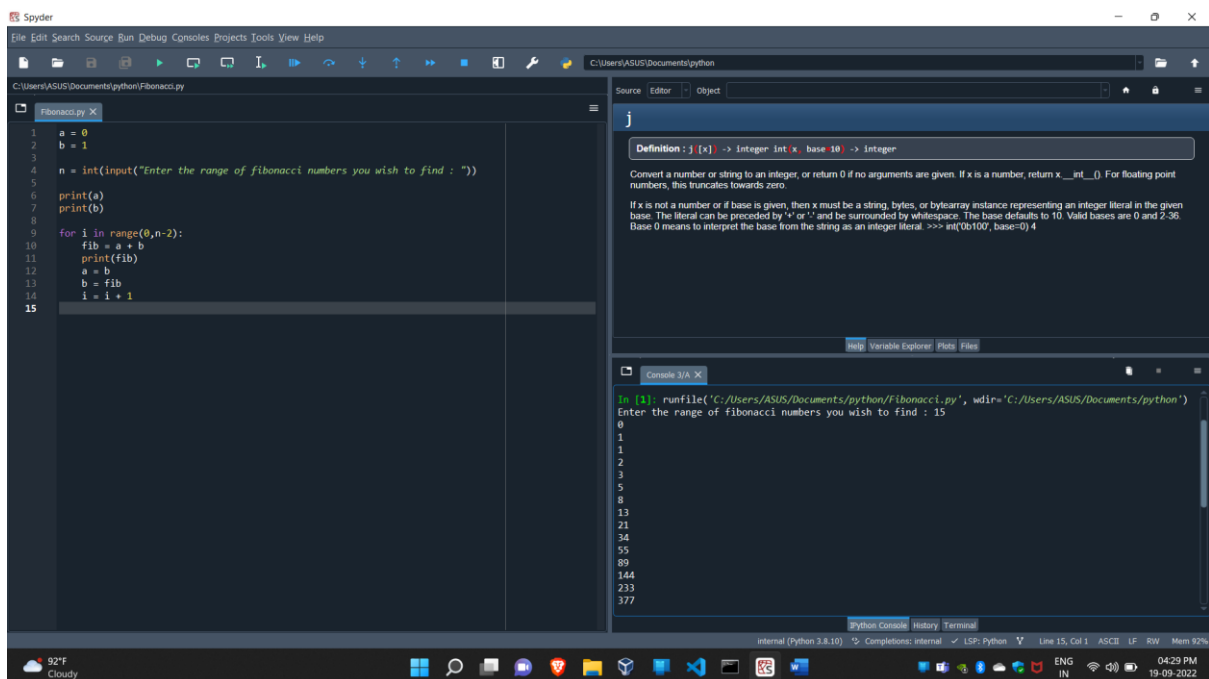
 print(fib)

 a = b

 b = fib

 i = i + 1

Output:



The screenshot shows the Spyder Python IDE interface. The editor window displays the following Python code for generating the Fibonacci series:

```
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
```

The console window shows the execution output:

```
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 : 15
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
```

The status bar at the bottom indicates the system is at 92°F, cloudy, and the time is 04:29 PM on 19-09-2022.