

IBM ASSIGNMENT 1:

1. Write a python program to test a given number is prime or not.

PROGRAM

```
n=int(input("enter the number"))

if n>1:

    for i in range(2,n):

        if(n%i)==0:

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

            break

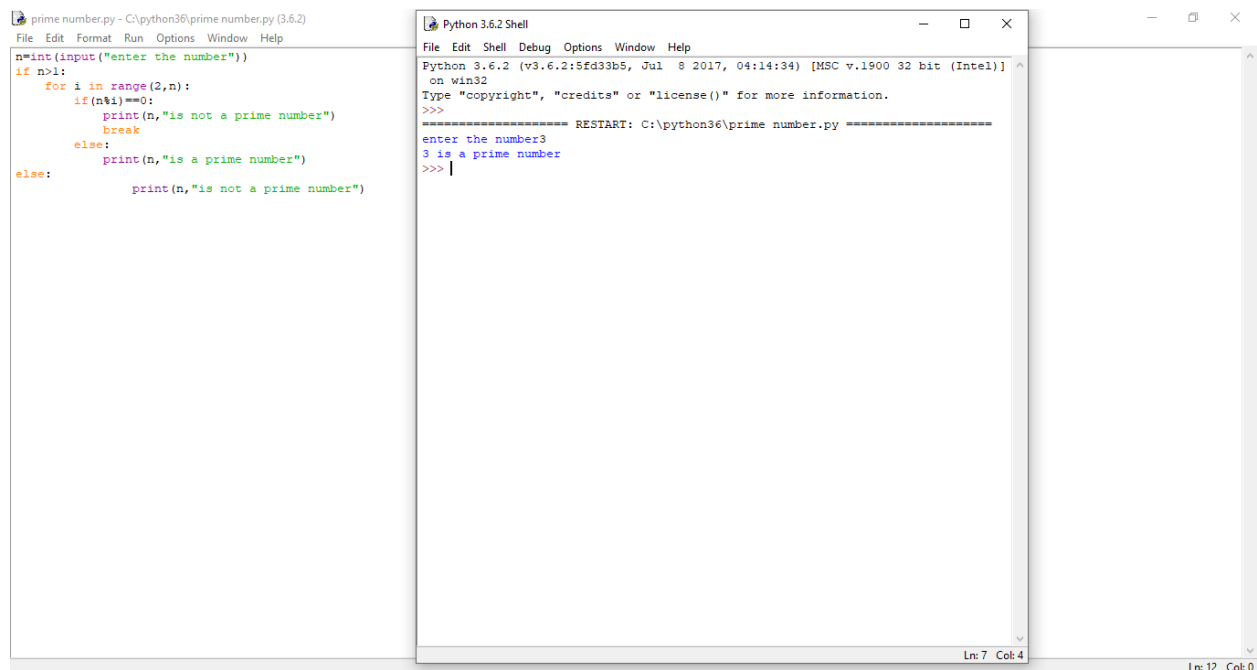
        else:

            print(n,"is a prime number")

    else:

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

OUTPUT



The screenshot displays two windows from a Python 3.6.2 IDE. The left window, titled 'prime number.py - C:\python36\prime number.py (3.6.2)', contains the source code for a prime number checker. The code prompts the user to enter a number, and if the number is greater than 1, it checks for divisibility from 2 to n-1. If any divisor is found, it prints that the number is not prime and breaks the loop. If no divisors are found, it prints that the number is prime. The right window, titled 'Python 3.6.2 Shell', shows the execution of the program. It displays the Python version and architecture, followed by a restart command. The user enters the number 3, and the program outputs '3 is a prime number'.

```
prime number.py - C:\python36\prime number.py (3.6.2)
File Edit Format Run Options Window Help
n=int(input("enter the number"))
if n>1:
    for i in range(2,n):
        if(n%i)==0:
            print(n,"is not a prime number")
            break
        else:
            print(n,"is a prime number")
    else:
        print(n,"is not a prime number")

Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python36\prime number.py =====
enter the number3
3 is a prime number
>>> |
```

2. Write a python program to display prime number series up to given numbers.

PROGRAM

```
first=int(input("enter first number"))
```

```
last=int(input("enter last number"))
```

```
for n in range(first,last+1):
```

```
    if n>1:
```

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

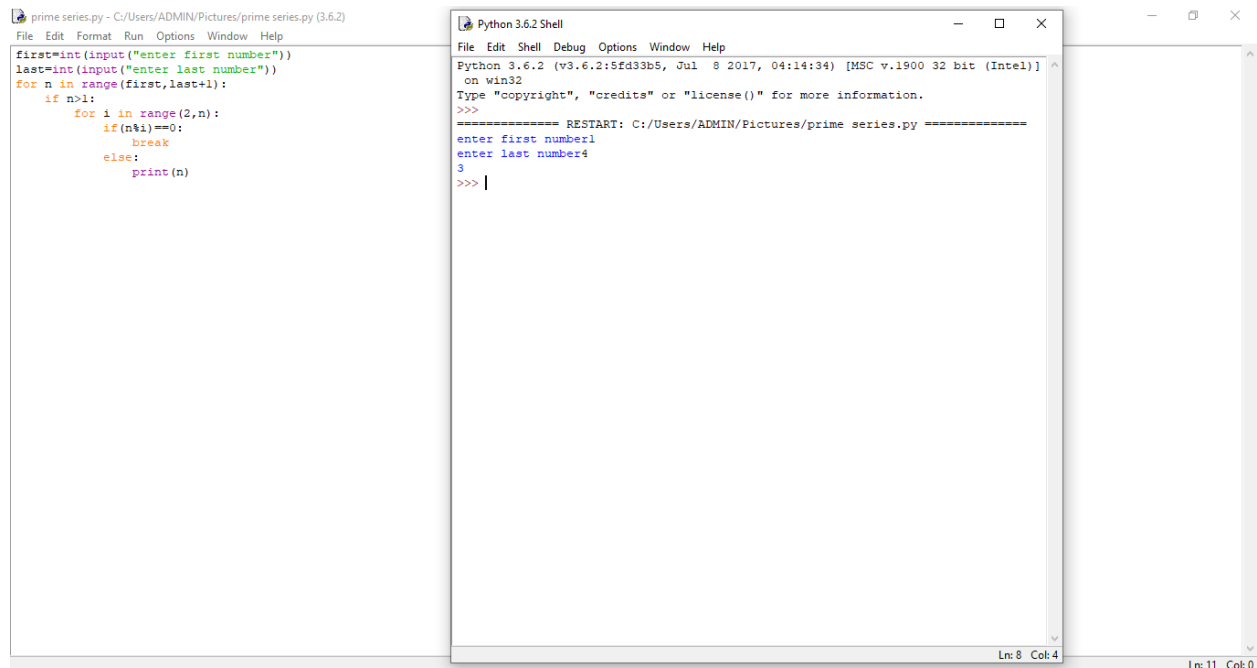
```
            if(n%i)==0:
```

```
                break
```

```
            else:
```

```
                print(n)
```

OUTPUT



```
prime series.py - C:/Users/ADMIN/Pictures/prime series.py (3.6.2)
File Edit Format Run Options Window Help
first=int(input("enter first number"))
last=int(input("enter last number"))
for n in range(first,last+1):
    if n>1:
        for i in range(2,n):
            if(n%i)==0:
                break
            else:
                print(n)

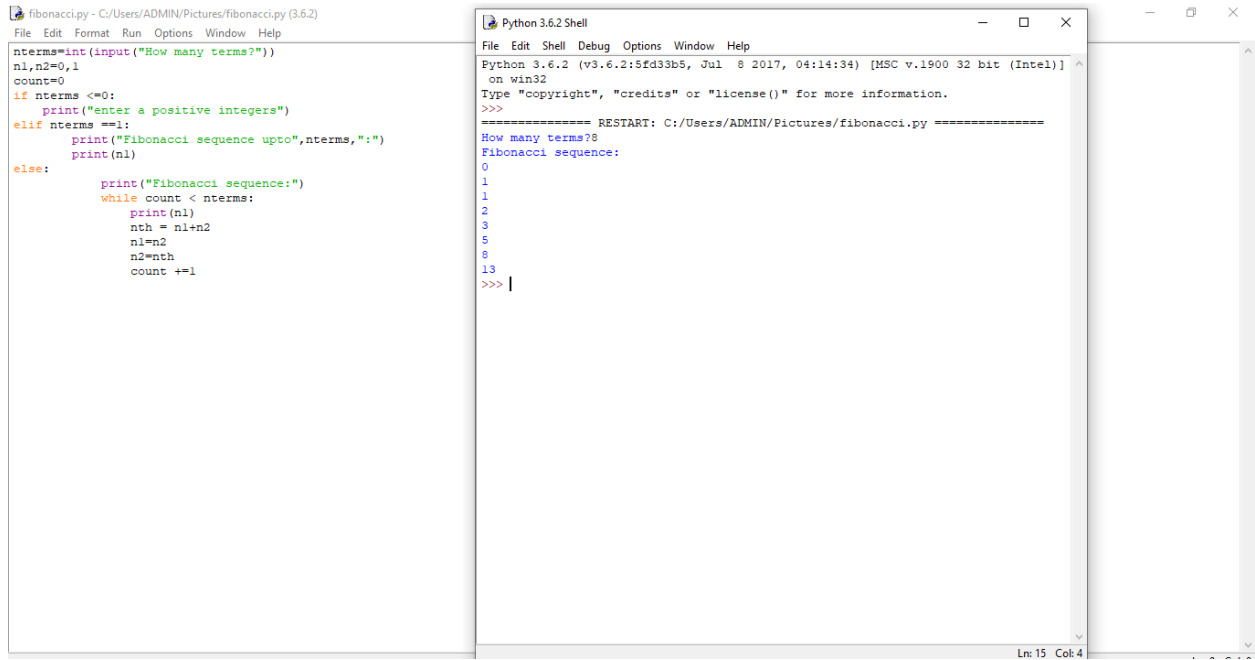
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ADMIN/Pictures/prime series.py =====
>>>
enter first number1
enter last number4
3
>>> |
```

3. Write a python program to generate Fibonacci series.

PROGRAM

```
nterms=int(input("How many terms?"))  
  
n1,n2=0,1  
  
count=0  
  
if nterms <=0:  
    print("enter a positive integers")  
  
elif nterms ==1:  
    print("Fibonacci sequence upto",nterms,":")  
    print(n1)  
  
else:  
    print("Fibonacci sequence:")  
    while count < nterms:  
        print(n1)  
        nth = n1+n2  
        n1=n2  
        n2=nth  
        count +=1
```

OUTPUT



The image shows two windows from a Python 3.6.2 IDE. The left window displays the source code for a program that generates a Fibonacci sequence. The right window shows the output of the program after execution.

```
fibonacci.py - C:/Users/ADMIN/Pictures/fibonacci.py (3.6.2)
File Edit Format Run Options Window Help
nterms=int(input("How many terms?"))
n1,n2=0,1
count=0
if nterms <=0:
    print("enter a positive integers")
elif nterms ==1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)
else:
    print("Fibonacci sequence:")
    while count < nterms:
        print(n1)
        nth = n1+n2
        n1=n2
        n2=nth
        count +=1

Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ADMIN/Pictures/fibonacci.py =====
How many terms?8
Fibonacci sequence:
0
1
1
2
3
5
8
13
>>> |
```

4. Write a program to generate odd numbers from m to n using while loop.

PROGRAM

```
maximum = int(input(" Please Enter the Maximum Value : "))
```

```
number = 1
```

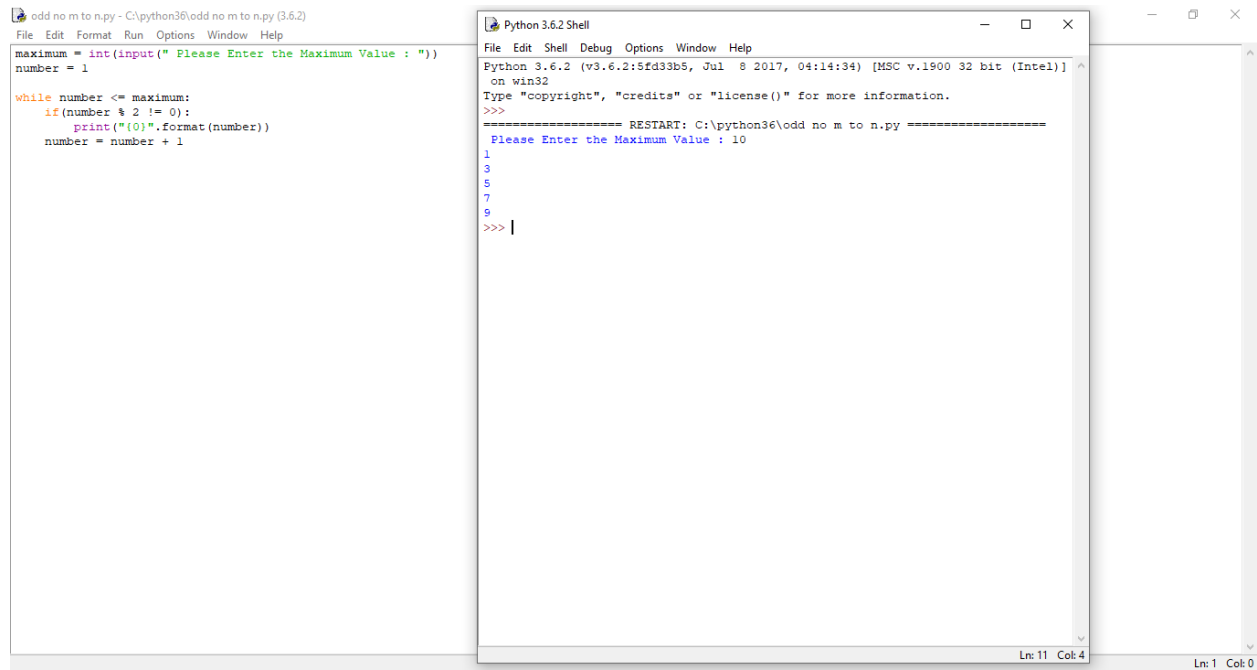
```
while number <= maximum:
```

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

```
        print("{0}".format(number))
```

```
    number = number + 1
```

OUTPUT



The image shows a screenshot of a Python IDE with two windows. The left window is a text editor titled 'odd no m to n.py - C:\python36\odd no m to n.py (3.6.2)'. It contains the following Python code:

```
maximum = int(input(" Please Enter the Maximum Value : "))
number = 1

while number <= maximum:
    if(number % 2 != 0):
        print("{0}".format(number))
    number = number + 1
```

The right window is a 'Python 3.6.2 Shell' titled 'Python 3.6.2 Shell'. It shows the execution output of the script. The output is as follows:

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python36\odd no m to n.py =====
Please Enter the Maximum Value : 10
1
3
5
7
9
>>> |
```

The status bar at the bottom of the shell window shows 'Ln: 11 Col: 4'.