

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

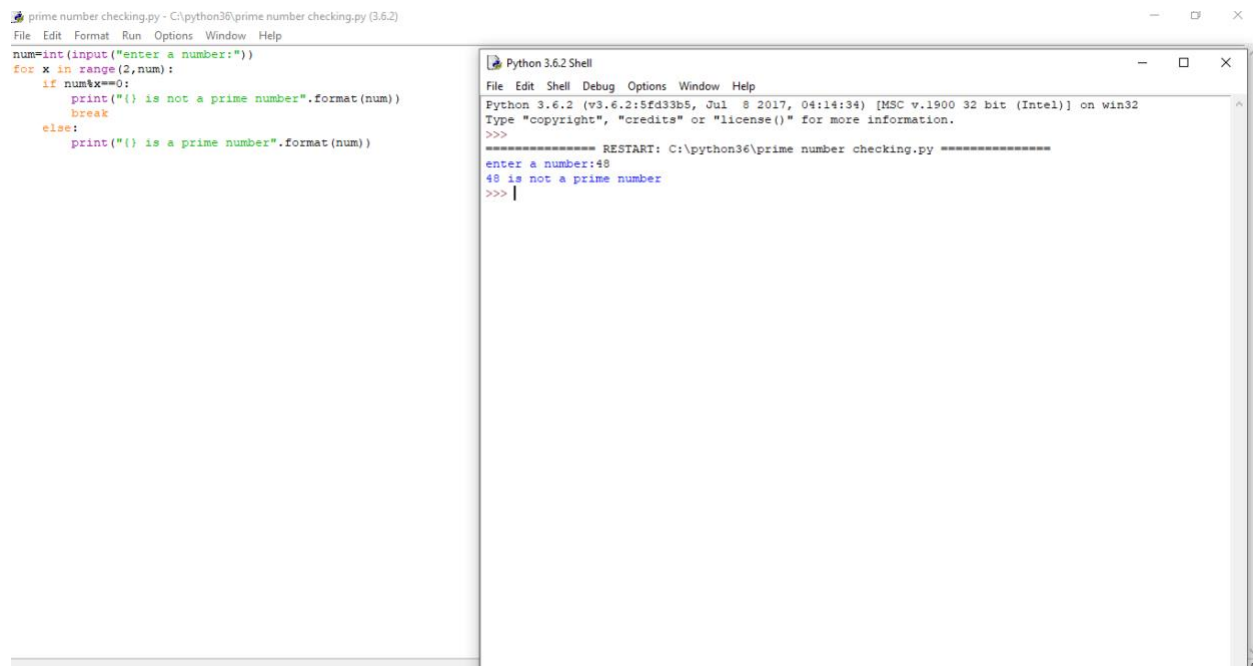
        else:

            print(n,"is a prime number")

    else:

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

OUTPUT



```
prime number checking.py - C:\python36\prime number checking.py (3.6.2)
File Edit Format Run Options Window Help
num=int(input("enter a number:"))
for x in range(2,num):
    if num%x==0:
        print("{} is not a prime number".format(num))
        break
    else:
        print("{} is a prime number".format(num))

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 checking.py =====
enter a number:48
48 is not 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



The screenshot displays a Python IDE with two windows. The left window, titled 'prime series.py - C:\python36\prime series.py (3.6.2)', contains the source code for a program that prints prime numbers between a first and last number. The code uses nested loops and a break statement to check for divisibility. The right window, titled 'Python 3.6.2 Shell', shows the execution of the program. It prompts the user to enter the first and last numbers, with '1' and '4' entered respectively. The output shows the prime numbers 2 and 3.

```
prime series.py - C:\python36\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)

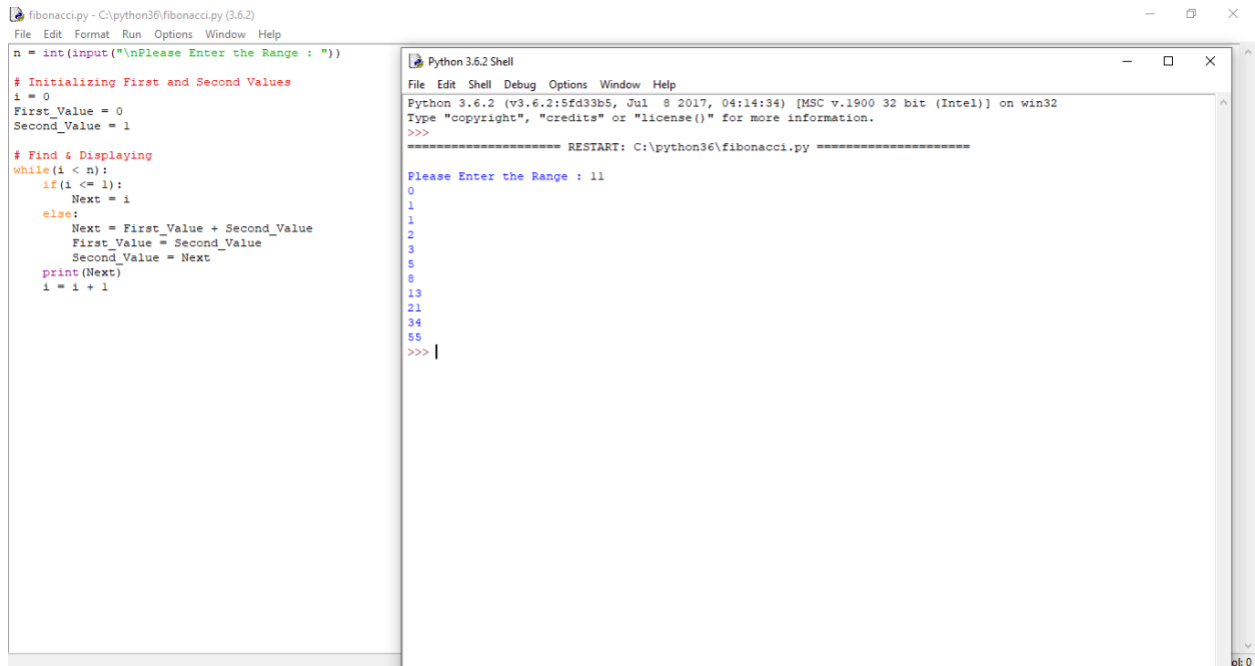
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python36\prime series.py =====
enter first number 1
enter last number 4
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 screenshot shows a Python 3.6.2 IDE with two windows. The left window, titled 'fibonacci.py - C:\python36\fibonacci.py (3.6.2)', contains the following code:

```
n = int(input("\nPlease Enter the Range : "))

# Initializing First and Second Values
i = 0
First_Value = 0
Second_Value = 1

# Find & Displaying
while(i < n):
    if(i <= 1):
        Next = i
    else:
        Next = First_Value + Second_Value
        First_Value = Second_Value
        Second_Value = Next
    print(Next)
    i = i + 1
```

The right window, titled 'Python 3.6.2 Shell', shows the output of the program. It displays the prompt 'Please Enter the Range : 11' followed by the Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55. The shell prompt '>>>' is visible at the bottom.

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
```

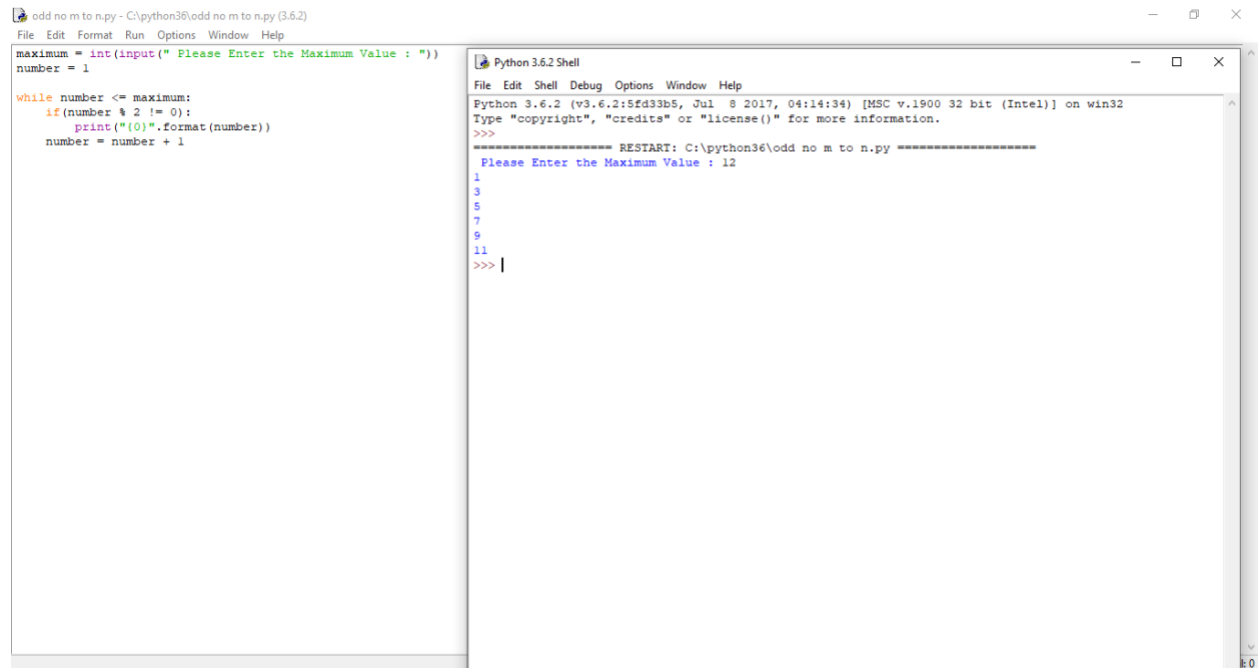
```
while number <= maximum:
```

```
    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 includes the version information, a restart message, and the list of odd numbers from 1 to 11, followed by a prompt for the next input.

```
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 : 12
1
3
5
7
9
11
>>> |
```