

**Assignment -4**  
**Python programming**

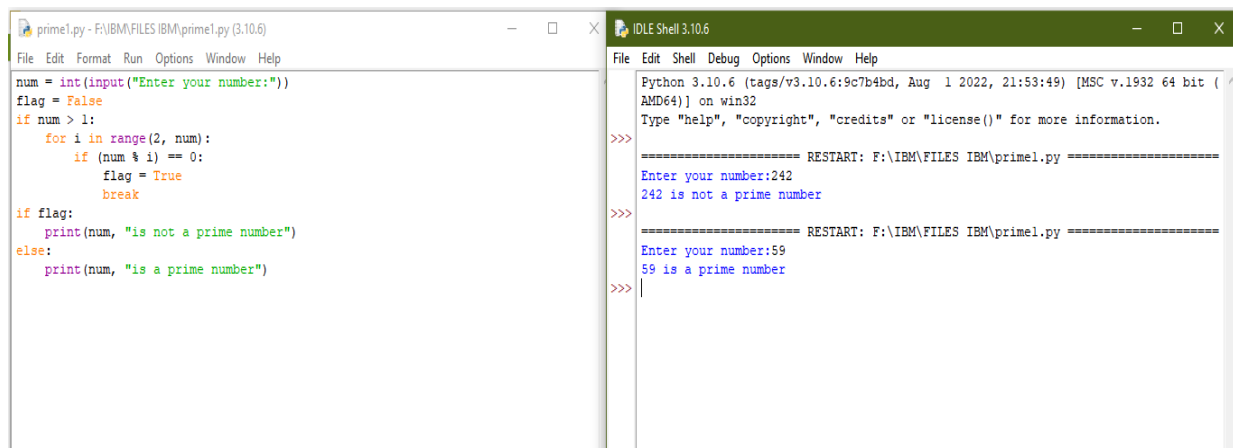
Assignment Date	19 September 2022
Student Name	Dhivya V
Student Roll Number	820419104019
Maximum Marks	2 Marks

**Question-1:**

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

**Solution:**

```
num = int(input("Enter your number:"))
flag = False
if num > 1:
    for i in range(2, num):
        if (num % i) == 0:
            flag = True
            break
if flag:
    print(num, "is not a prime number")
else:
    print(num, "is a prime number")
```

The screenshot shows two windows from a Python IDE. The left window, titled 'prime1.py - F:\IBM\FILES IBM\prime1.py (3.10.6)', contains the Python code for checking if a number is prime. The code uses a flag variable and a loop to test divisibility. The right window, titled 'IDLE Shell 3.10.6', shows the execution of the program. It displays the prompt 'Enter your number:242' followed by the output '242 is not a prime number'. Below this, it shows another execution with 'Enter your number:59' and the output '59 is a prime number'. The shell window also shows the restart command and the Python version information.

**Question-2:**

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

**Solution:**

```
start= int(input(" Please Enter A minimum value:"))
end=int(input(" Please Enter B Maximum Value : "))
for num in range(start, end + 1):
    if num % 2 != 0:
        print(num, end = " ")
```

The screenshot shows an IDE with two windows. The left window, titled 'odd.py - F:\IBM\FILES IBM\odd.py (3.10.6)', contains the following Python code:

```
start= int(input(" Please Enter A minimum value:"))
end=int(input(" Please Enter B Maximum Value : "))
for num in range(start, end + 1):
    if num % 2 != 0:
        print(num, end = " ")
```

The right window, titled 'IDLE Shell 3.10.6', shows the execution output:

```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\IBM\FILES IBM\odd.py =====
Please Enter A minimum value:367
Please Enter B Maximum Value : 400
367 369 371 373 375 377 379 381 383 385 387 389 391 393 395 397 399
>>>
```

### Question-3:

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

#### Solution:

```
lower =int(input(" Please Enter minimum value:"))
upper =int(input(" Please Enter maximum value:"))

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num)
```

The screenshot shows an IDE with two windows. The left window, titled 'prime series.py - F:\IBM\FILES IBM\prime series.py (3.10.6)', contains the following Python code:

```
lower =int(input(" Please Enter minimum value:"))
upper =int(input(" Please Enter maximum value:"))

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num)
```

The right window, titled 'IDLE Shell 3.10.6', shows the execution output:

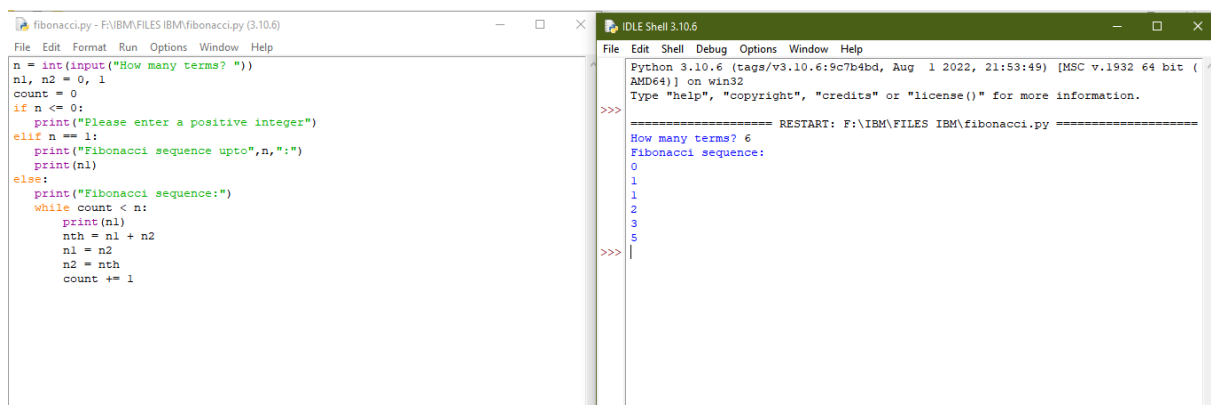
```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\IBM\FILES IBM\prime series.py =====
Please Enter minimum value:100
Please Enter maximum value:200
Prime numbers between 100 and 200 are:
101
103
107
109
113
127
131
137
139
149
151
157
163
167
173
179
181
191
193
197
199
>>>
```

#### Question-4:

Write a python program to generate Fibonacci series

#### Solution:

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



The screenshot displays two windows from a Python IDE. The left window, titled 'fibonacci.py - F:\IBM\FILES IBM\fibonacci.py (3.10.6)', contains the Python code for generating the Fibonacci series. The code prompts the user for the number of terms, handles non-positive inputs, and uses a while loop to calculate and print the sequence. The right window, titled 'IDLE Shell 3.10.6', shows the program's execution. It displays the prompt 'How many terms? 6', the output 'Fibonacci sequence:', and the resulting sequence of numbers: 0, 1, 1, 2, 3, 5.

```
fibonacci.py - F:\IBM\FILES IBM\fibonacci.py (3.10.6)
File Edit Format Run Options Window Help
n = int(input("How many terms? "))
n1, n2 = 0, 1
count = 0
if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print("Fibonacci sequence upto",n,":")
    print(n1)
else:
    print("Fibonacci sequence:")
    while count < n:
        print(n1)
        nth = n1 + n2
        n1 = n2
        n2 = nth
        count += 1

IDLE Shell 3.10.6
File Edit Shell Debug Options Window Help
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\IBM\FILES IBM\fibonacci.py =====
How many terms? 6
Fibonacci sequence:
0
1
1
2
3
5
>>>
```