**10b** Write a python function for prime number input limit in asparameter to it.

**Objectives:**

1. To learn about python as scripting option.

**Theory:**

**Prime Number:**

A prime number is a whole number greater than 1 whose only factors are 1 and itself. A factor is a whole numbers that can be divided evenly into another number. The first few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23 and 29.

**How do for loops work?**

Many languages have conditions in the syntax of their **for** loop, such as a relational expression to determine if the loop is done, and an increment expression to determine the next loop value. In Python this is controlled instead by generating the appropriate sequence. Basically, any object with an iterable method can be used in a **for** loop. Even strings, despite not having an iterable method - but we'll not get on to that here. Having an iterable method basically means that the data can be presented in list form, where there are multiple values in an orderly fashion. You can define your own iterables by creating an object with next() and iter() methods.

**Nested loops:**

When you have a block of code you want to run **x** number of times, then a block of code within that code which you want to run **y** number of times, you use what is known as a "nested loop". In Python, these are heavily used whenever someone has a list of lists - an iterable object within an iterable object.

for x in xrange(1, 11):

for y in xrange(1, 11):

print '%d \* %d = %d' % (x, y, x\*y)

**Early Exits:**

Like the while loop, the for loop can be made to exit before the given object is finished. This is done using the break statement, which will immediately drop out of the loop and contine execution at the first statement after the block. You can also have an optional else clause, which will run should the for loop exit cleanly - that is, without breaking.

for x in xrange(3):

if x == 1:

break

**Program:**

n=int(input("Enter number:"))

for i in range(1,n+1):

c=0

* print(i)
* print()

if(i==1): continue if(i==2): print(2) continue

for j in range(2,i-1): if(i%j==0): c=c+1

if(c==0):

print(i)

**Output:**

sarita@HP-Laptop-15g-dr0xxx:~$ python 10b.py

Enter number:20

2

3

5

7

11

13

17

19

**Conclusion:**

1.Basics of python like the concept of loops learnt.

2. Conditional statements learnt.

**References:**

[1] https://docs.python.org/3/