|  |  |
| --- | --- |
| **Ex.No:8** | **Finding N Primes** |
|  |

***Aim:***

To develop a python program that generates N numbers of primes from number 1.

***Main Algorithm:***

|  |  |
| --- | --- |
| **Step 1:** | Start Process |
| **Step 2:** | Get limit from user |
| **Step 3:** | Assign i as 1 and count as 0 |
| **Step 4:** | Call i with prime\_finder |
| **Step 5:** | If i is prime number print i and increment count |
| **Step 6:** | Increment i |
| **Step 7:** | If count is equals to limit then goto Step 9 |
| **Step 8:** | Else goto Step 4 |
| **Step 9** | Stop Process |

***Prime Finder Algorithm:***

|  |  |
| --- | --- |
| **Step 1:** | Start Process |
| **Step 2:** | Receive number |
| **Step 3:** | Assign is\_prime as True |
| **Step 4:** | If number is 1 or 2 or 3 then return is\_prime and goto Step 11 |
| **Step 5:** | Else assign i with 2 |
| **Step 6:** | If number % i is equals to 0 then assign is\_prime as False and goto Step 10 |
| **Step 7:** | Else Increment i |
| **Step 8:** | If i is equals to number then goto 10 |
| **Step 9:** | Else goto Step 6 |
| **Step 10:** | Return is\_prime |
| **Step 11:** | Stop Process |

**Flow Chart:**

**Pseudo Code Prime Finder:**

START

RECIVE number

ASSIGN is\_prime = True

ASSIGN i=2

IF i=1 OR i=2 OR i=3 THEN

RETURN is\_prime

ELSE

WHILE i is less than number -1:

IF number % i is equals to 0 THEN

ASSIGN is\_prime = False

BREAK

END IF

INCREMENT i

END WHILE

END IF

RETURN is\_prime

STOP

**Pseudo Code Main Algorithm:**

START

READ limit

ASSIGN count = 0

ASSIGN i = 1

WHILE count is not equals to limit THEN

is\_prime = Call Prime Finder with i

IF is\_prime is equals to True THEN

PRINT i

INCREMENT count

END IF

INCREMENT i

END WHILE

STOP

**Program:**

def prime\_finder(number):

is\_prime = True

if (number==1 or number==2 or number==3):

return is\_prime

else:

for i in range (2,number):

if(number%i == 0):

is\_prime = False

break

return is\_prime

print ("Finding N Prime numbers")

print ("-----------------------")

limit = int(input("Enter limit:"))

count = 0

i = 1

while(count != limit):

is\_prime = prime\_finder(i)

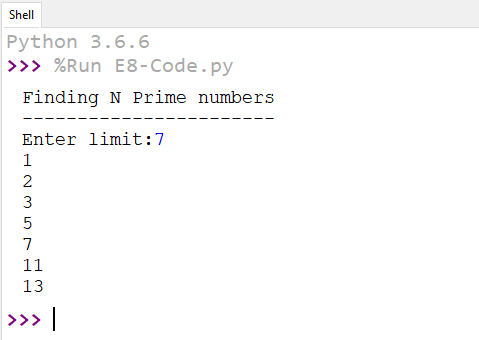
if(is\_prime):

print(i)

count+=1

i+=1

**Output:**

****

***Result:***

Thus the program that generates N numbers of primes from number 1 was developed and tested successfully.