Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Фізико-технічний інститут

Кафедра фізико-технічних засобів захисту інформації

**Протокол лабораторної роботи № 3**

з дисципліни Програмування 4

за темою

Принципи програмування на мові Python

**Виконав:**

Студентка 2 курсу

Групи ФЕ-71

Христюк І.В.

**Прийняв:**

Доц. Прогонов Д.О.

**Київ 2019**

**Мета роботи**: Отримати навички роботи з циклами на мові Python

## Порядок виконання роботи

1. Знайти суму ряду з точністю ε=10-4, загальний член якого



**Використовуючи оператор циклу for, розв’язати наступні задачі:**

Дано натуральне число *n*. Знайти всі числа Мерсена, що не перевищують число *n*. Просте число називається *числом Мерсена*, якщо його можна представити у вигляді , де  – теж просте число.

**Реалізація алгоритму**

import math  
def f(n):  
 return (1/(2\*\*n)+1/(3\*\*n))  
  
e = 0.0001  
seqSum = 0.0  
n = 1  
while f(n) >= e:  
 seqSum += f(n)  
 print("{0} element your sequence: {1}".format(n, f(n)))  
 print("Sum sequence accuracy 0.0001: {0}".format(seqSum))  
 n += 1  
print("Sum sequence accuracy 0.0001: {0}".format(seqSum))

import math  
def f(n):  
 return (1/(2\*\*n)+1/(3\*\*n))  
e = 0.0001  
seqSum = 0.0  
n = 1  
while True:  
 if (f(n) >= e):  
 seqSum += f(n)  
 print("{0} element your sequence: {1}".format(n, f(n)))  
 n += 1  
 else:  
 break  
print("Sum sequence accuracy 0.0001: {0}".format(seqSum))

from math import sqrt  
  
RANGE\_PRIME\_NUMBERS = 50  
  
def prime\_number(): #простые числа  
 generated\_number = [2]  
 for p in range(3, RANGE\_PRIME\_NUMBERS+1, 2):  
 if p > 10 and p % 10 == 5:  
 continue  
 for k in generated\_number:  
 if k > int((sqrt(p))+1):  
 generated\_number.append(p)  
 break  
 if p % k == 0:  
 break  
 else:  
 generated\_number.append(p)  
 return generated\_number  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 numbers = []  
 while True:  
 try:  
 n = int(input('ENTERED YOUR N: '))  
 break  
 except ValueError:  
 print("Value error!")  
 for p in prime\_number():  
 temp = 2 \*\* p - 1  
 if temp < n:  
 numbers.append(temp)  
 else:  
 break  
print(numbers)