Міністерство освіти і науки України

НТУУ «Київський політехнічний інститут»

Фізико-технічний інститут

# Програмування 4

# Лабораторна робота №5

«Масиви у мові Python»

**Виконав:**

Студент II курсу ФТІ групи ФЕ-81

Юрченко Денис

2020

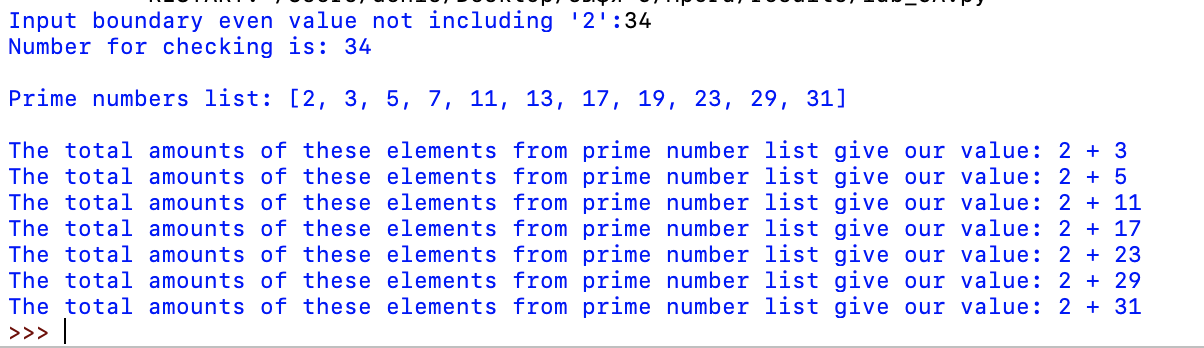
1. Завдання лабораторної роботи(А)

Дано парне число ; перевірити для цього гіпотезу Гольбаха. Ця гіпотеза (на сьогоднішній день не спростована та повністю не доведена) полягає в тому, що кожне парне число , що більше 2, можна представити у вигляді суми двох простих чисел. (Визначити функцію, що дозволяє розпізнавати прості числа.)

1. Код реалізації

|  |
| --- |
| Lab\_5А.py |
| while True:  try:  N = float(input("Input boundary even value not including '2':"))  if N == "nan":  print("Can‘t be Not a Number")  continue  elif N == float('inf'):  print("Infinity not excepted!")  continue  n = float(N)  if n - int(n) == 0:  n = int(n)  if n % 2 != 0:  print("Please, input even number")  continue  elif n < 0:  print("Positive even numbers only!")  continue  elif n == 2:  print("Please, input another even number")  continue  elif n == 0:  print("Please, input another even number")  continue  break  else:  continue    except ValueError:  print("Invalid value entered!")  def prime\_check(n):  print("Number for checking is:", n, '\n')  numbers = list(range(2, n + 1))  for i in numbers:  if i != 0:  for k in range(2 \* i, n + 1, i):  numbers[k - 2] = 0  primes = [x for x in numbers if x != 0]  print("Prime numbers list:", primes, '\n')  def task():  for position in range(len(primes)):  z = n - primes[position]  if z in primes:  print("The total amounts of these elements from prime number list give our value:", 2, "+",  primes[position])  position += 1  task()  if \_\_name\_\_ == '\_\_main\_\_':  prime\_check(n) |

1. Виконання програми



1. Завдання лабораторної роботи(В)

Піднести до додатного цілого степеня дійсне ненульове число Аналіз умови

1. Код реалізації

|  |
| --- |
| Lab\_5В.py |
| import math  def if\_nan():  print("Error! Not a number!")  def fast\_exp(b, n):  def even(n):  if n % 2 == 0:  return 1  return 0  if n == 0:  return 1  if even(n):  return fast\_exp(b, n / 2) \*\* 2  return b \* fast\_exp(b, n - 1)  while True:  try:  B = input("Input any rational number, please:")  if B == "pi":  b = math.pi  break  elif B == "e":  b = math.e  break  elif B == "inf":  b = float('inf')  elif B == "nan":  if\_nan()  continue  b = float(B)  break  except valueError:  print("Invalid value entered!")  while True:  try:  n = float(input("Input value‘s positive integer power:"))  if not n > 0 or n % 1 != 0:  print("Error! value‘s power has to be positive and integer. Please, input the proper one:")  continue  break  except ValueError:  print("Invalid value enteredt")  if \_\_name\_\_ == '\_\_main\_\_':  print(fast\_exp(b, n)) |

1. Виконання програми

Изображение выглядит как нож

Автоматически созданное описание