Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчёт**

по третьей лабораторной работе

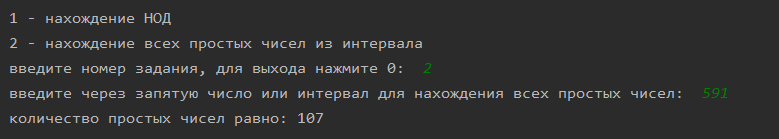
Студент: Пицуха Я. А.

ФИТ 3 курс 4 группа

Минск 2020

1. Используя собственное приложение, найти все простые числа в интервале [2, n]. Значение n соответствует варианту из таблицы, указанному преподавателем.

Вариант 11, n = 591

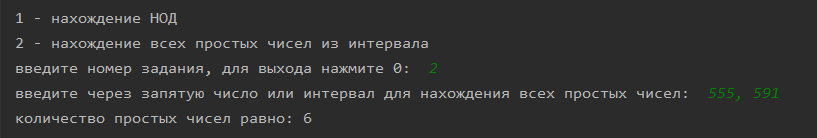


Ответ: 107

Проверяем по свойству простых чисел наше число n / ln(n) = 92.607

1. Повторить п.1 для интервала [m, n].

m = 555, n = 591



Ответ: 6

1. Записать числа m и n в виде произведения простых множителей (форма записи – каноническая).

m = 555 = 3 \* 5 \* 37

n = 591 = 3 \* 197

1. Проверить, является ли число, состоящее из конкатенации цифр m и n простым.

m = 555, n = 591

число 555591

корень этого числа округлённый в большую сторону равен 745.380

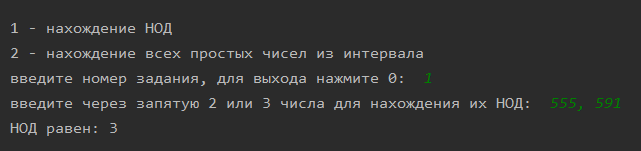
проверяем все простые числа от 2 до√555591

сумма цифр данного числа делится на 3, следовательно и наше число делится на 3 и оно составное

Ответ: не является

1. Найти НОД (m, n).

m = 555, n = 591



Ответ: 3

Листинги приложения:

from lab3 import findNOD, primeSearch  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 while True:  
 print()  
 print('1 - нахождение НОД ')  
 print('2 - нахождение всех простых чисел из интервала ')  
 print('введите номер задания, для выхода нажмите 0: ', end=' ')  
 inp = int(input())  
 if inp == 0:  
 break  
 elif inp == 1:  
 print('введите через запятую 2 или 3 числа для нахождения их НОД: ', end=' ')  
 inpnod = str(input()).split(',')  
 if len(inpnod) == 2:  
 print(f'НОД равен: {findNOD.findNOD(int(inpnod[0]), int(inpnod[1]))}')  
 elif len(inpnod) == 3:  
 print(f'НОД равен: {findNOD.findNOD(int(inpnod[0]), int(inpnod[1]), int(inpnod[2]))}')  
 else:  
 print('неверное количество чисел')  
 elif inp == 2:  
 print('введите через запятую число или интервал для нахождения всех простых чисел: ', end=' ')  
 inpprime = str(input()).split(',')  
 if len(inpprime) == 1:  
 print(f'количество простых чисел равно: {len(primeSearch.primeSearch(int(inpprime[0])))}')  
 elif len(inpprime) == 2:  
 print(f'количество простых чисел равно: {len(primeSearch.primeSearch(int(inpprime[1]), int(inpprime[0])))}')  
 else:  
 print('неверное количество чисел')  
 else:  
 print('такого номера нет')

def findNOD(a, b, c=None):  
 *"""  
 нахождение НОД чисел* ***:param*** *a: первое число* ***:param*** *b: второе число* ***:param*** *c: третье число* ***:return****: НОД  
 """* while True:  
 a, b = b, (a - (a // b) \* b)  
 if b == 0:  
 if c is not None:  
 return findNOD(c, a)  
 else:  
 return a

import math  
  
  
def primeSearch(n, m=2):  
 *"""  
 функция для нахождения количества простых чисел на интервале* ***:param*** *n: конец интервала* ***:param*** *m: начало интервала* ***:return****: количество простых чисел  
 """* if m == 2:  
 primesList = [i for i in range(m, n + 1)]  
 for it, vt in enumerate(primesList):  
 for ib, vb in enumerate(primesList):  
 if vt is not vb and vb % vt == 0:  
 del primesList[ib]  
 else:  
 continue  
 return primesList  
 else:  
 sq = round(math.sqrt(n))  
 primesList2 = primeSearch(sq)  
 primesList = [i for i in range(m, n + 1)]  
 for it, vt in enumerate(primesList2):  
 for ib, vb in enumerate(primesList):  
 if vt is not vb and vb % vt == 0:  
 del primesList[ib]  
 else:  
 continue  
 return primesList

