**[Проект Эйлера](https://euler.jakumo.org/)**

**asdarov**

**qwerty1234**

**Задача 1**

**Числа, кратные 3 или 5**

Если выписать все натуральные числа меньше 10, кратные 3 или 5, то получим 3, 5, 6 и 9. Сумма этих чисел равна 23.

Найдите сумму всех чисел меньше 1000, кратных 3 или 5.

**Натуральные числа** (от [лат.](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *naturalis* — естественный) — [числа](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE), возникающие естественным образом при счёте (например, 1, 2, 3, 4, …). Последовательность всех натуральных чисел, расположенных в порядке возрастания, называется **натуральным рядом**.

a = []

**for** i **in** range(1,1000):

if i % 3 == 0 or i % 5 == 0:

a.append(i)

sum(a)

ответ: 233168

**или**

a = [i **for** i **in** range(30,250) **if** i%30 == 0 **or** i%31 == 0]

**Задача 2**

**Четные числа Фибоначчи**

Каждый следующий элемент ряда Фибоначчи получается при сложении двух предыдущих. Начиная с 1 и 2, первые 10 элементов будут:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Найдите сумму всех четных элементов ряда Фибоначчи, которые не превышают четыре миллиона.

F0=0; F1=1; Fn=F(n-1)+F(n-2) ; n>=2 Это общая формула

fibo = [0,1]

for index in range (2, 100):

fibo.append(fibo[index-1]+fibo[index-2])

fibo = [x for x in fibo if x % 2 == 0 and x<4000000]

sum(fibo)

**Ответ 4613732**

Плохое решение. Список заранее известной длины, хотя по условию несколько иначе

def fibonacci(max):        # генератор (а не функция, т.к. оператор return заменён на yield)  
    a, b = 0, 1  
    while a < max:  
        yield a            # return a, + запоминаем место рестарта для следующего вызова  
        a, b = b, a + b    # параллельное присваивание, которое выполняется одновременно и параллельно  
  
for n in fibonacci(4000000):   # используем генератор fibonacci() как итератор  
    print (n)               # печатаем все числа Фибоначчи меньшие 100 через пробел

итог

fibo = []

def fibonacci(max):

a, b = 0, 1

while a < max:

yield a

a, b = b, a + b

for n in fibonacci(4000000):

if n % 2 == 0:

fibo.append(n)

sum(fibo)

**Задача 3**

**Наибольший простой делитель**

Простые делители числа 13195 - это 5, 7, 13 и 29.

Каков самый большой делитель числа 600851475143, являющийся простым числом?

Итак, У каждого натурального числа, большего единицы, имеются по крайней мере два натуральных делителя: единица и само это число. При этом **натуральные числа, имеющие ровно два делителя**, называются простыми, а имеющие больше двух делителей — составными. Единица имеет ровно один делитель и не является ни простым, ни составным.   
У каждого натурального числа, большего 1, есть хотя бы один простой делитель.

**Просто́е число́** — это натуральное **число**, больше единицы, имеющее ровно два натуральных делителя: 1 и само себя.

<https://www.youtube.com/watch?v=ygA8EMHbtxU>

#import math

n=600851475143

i=1

a=[]

while i\*i<=n:

if n % i ==0:

if i==n//i:

a.append(i)

else:

a.append(i)

a.append(n//i)

i=i+1

a.sort()

print(a)

def isprime(m):

for ii in range (2,m-1):

if m % ii == 0:

# print('делитель на' ,ii , 'составное')

res=1

return res

else:

ii=ii+1

res=0

return res

for iii in range (0,len(a)):

m=a[iii]

isprime(m)

print(isprime(m))

if isprime(m)==0:

print('число простое', m)

if isprime(m)==1:

print('число составное', m)

iii=iii+1

Это вывод всех делителей. Дальше в полученноом списке проверить на простоту . Простой делится без остатка только на себя и на 1.

Index in range (0,len(a))

Можно и нужно оптимизировать код,не эффективный. Хотя бы поиск сузить

ОТВЕТ 6857

**Задача 4**

**Наибольшее произведение-палиндром**

Число-палиндром с обеих сторон (справа налево и слева направо) читается одинаково. Самое большое число-палиндром, полученное умножением двух двузначных чисел – 9009 = 91 × 99.

Найдите самый большой палиндром, полученный умножением двух трехзначных чисел.

def reverse(a):

b=str(a)

int(b[::-1])

return int(b[::-1])

m=[]

for i in range (100,1000):

# print(i)

for y in range (100,1000):

# print(y\*i)

m.append(y\*i)

m.sort(reverse=True)

max=0

for z in range (0,len(m)):

if m[z]==reverse(m[z]):

if m[z] > max:

max=m[z]

print(m[z])

exit

else:

z+=1

Тупой перебор. Хорошо бы оптимизировать

Ответ: 906609

**Задача 5**

**Наименьшее кратное**

2520 - самое маленькое число, которое делится без остатка на все числа от 1 до 10.

Какое самое маленькое число *делится нацело* на все числа от 1 до 20?

def proverka(n):

for i in range(11,21):

if n % i ==0:

# print(i, 'почему')

# i=i+1

res=1

else:

res=0

break

return res

x=2520

while proverka(x)!=1:

x=x+1

print(x)

Идея в том ,что берем перебор от 11 до 20,ибо они уже в себя включают первый 10(кроме 11,13,17,19). Долгий тупой перебор,надо оптимизировать  
{\displaystyle F\_{0}=0,\quad F\_{1}=1,\quad F\_{n}=F\_{n-1}+F\_{n-2},\ n\geqslant 2,\ n\in \mathbb {Z} .}

**Ответ 232792560**

**Задача 6**

**Разность между суммой квадратов и квадратом суммы**

Сумма квадратов первых десяти натуральных чисел равна

12 + 22 + ... + 102 = 385

Квадрат суммы первых десяти натуральных чисел равен

(1 + 2 + ... + 10)2 = 552 = 3025

Следовательно, разность между суммой квадратов и квадратом суммы первых десяти натуральных чисел составляет 3025 − 385 = 2640.

Найдите разность между суммой квадратов и квадратом суммы первых ста натуральных чисел.

n=0

m=0

for i in range (1,101):

print(i, 'натуральное')

# i=i\*i

print(i\*i, 'квадрат')

n=n+i\*i

m=m+i

print(n,'сумма квадратов')

print(m\*m, 'квадрат суммы')

print (m\*m-n, 'разность квадрата суммы и суммы квадратов')

**Ответ**

**338350 сумма квадратов**

**25502500 квадрат суммы**

**25164150 разность квадрата суммы и суммы квадратов**

**Задача 7**

**10001-ое простое число**

Выписав первые шесть простых чисел, получим 2, 3, 5, 7, 11 и 13. Очевидно, что 6-ое простое число - 13.

Какое число является 10001-ым простым числом?

Использовал в реализации тест являющийся продолжением малой теоремы Ферма. Берем проверяемое число **number** и 20 любых чисел **p**(случайных) меньше number. Если p\*\*(number-1) % number = 1, то число простое . То есть a \*\*(p-1) mod p =1. Это вероятностный тест

<https://www.youtube.com/watch?v=z5ZtFJiYJaU>

<https://www.youtube.com/watch?v=ASIbwazJFhQ>

<http://www.cyberforum.ru/cpp-beginners/thread660361.html>

Считает долго.

Ответ: 104743.

Список простых чисел можно взять на странице <http://unconnected.info/sn10.aspx>

**Задача 8**

**Наибольшее произведение в последовательности**

Наибольшее произведение четырех последовательных цифр в нижеприведенном 1000-значном числе равно 9 × 9 × 8 × 9 = 5832.

73167176531330624919225119674426574742355349194934  
96983520312774506326239578318016984801869478851843  
85861560789112949495459501737958331952853208805511  
12540698747158523863050715693290963295227443043557  
66896648950445244523161731856403098711121722383113  
62229893423380308135336276614282806444486645238749  
30358907296290491560440772390713810515859307960866  
70172427121883998797908792274921901699720888093776  
65727333001053367881220235421809751254540594752243  
52584907711670556013604839586446706324415722155397  
53697817977846174064955149290862569321978468622482  
83972241375657056057490261407972968652414535100474  
82166370484403199890008895243450658541227588666881  
16427171479924442928230863465674813919123162824586  
17866458359124566529476545682848912883142607690042  
24219022671055626321111109370544217506941658960408  
07198403850962455444362981230987879927244284909188  
84580156166097919133875499200524063689912560717606  
05886116467109405077541002256983155200055935729725  
71636269561882670428252483600823257530420752963450

Найдите наибольшее произведение тринадцати последовательных цифр в данном числе

Использовал запись такого большого числа через ‘’’ ‘’’ (тройной апостроф), и затем функцию

a=a.replace("\n","") (замену,убирал перенос на другую строку)

Разбил данную строку на 13 символов, и передавал в функцию, которая возвращала произведение

Ответ 23514624000