<https://eu8.proxysite.com/>

<https://www.e-olymp.com/uk/problems/46>

Відрізки

Розріжте відрізок довжиною **L** на найбільшу кількість частин, довжини яких – натуральні числа, щоб з них не можна було скласти жодного трикутника.

**Вхідні дані**

Довжина відрізку **L**. **3** < **L** < **2000000000**.

**Вихідні дані**

Шукана кількість частин.

Ліміт часу **1** секунда

Ліміт використання пам'яті **122.17** MiB

**Вхідні дані #1**

6

**Вихідні дані #1**

3

n=int(input())

k=2

f1=1

f2=2

s=f1+f2

while s<n:

 f3=f1+f2

 s=s+f3

 k=k+1

 f1=f2

 f2=f3

print(k)

<https://www.e-olymp.com/uk/problems/4730>

Фібоначчі

Числа Фібоначчі - це послідовність чисел F(**n**), яка задається формулою:

F(**0**) = **1**, F(**1**) = **1**, F(**n**) = F(**n** - **1**) + F(**n** - **2**)

За заданим числом **n** вивести **n**-те число Фібоначчі.

**Вхідні дані**

Невід'ємне число **n** (**n** ≤ **45**) - номер числа Фібоначчі, яке потрібно вивести.

**Вихідні дані**

Вивести **n**-те число Фібоначчі.

Ліміт часу **1** секунда

Ліміт використання пам'яті **128** MiB

**Вхідні дані #1**

4

**Вихідні дані #1**

5

n=**int**(input())

a=**dict**()

a[1]=1

a[2]=1

**for** i **in** **range**(3,n+2):

 a[i]=a[i-1]+a[i-2]

**print**(a[n+1])

<https://www.e-olymp.com/uk/problems/8960>

Крім найменших і найбільших

З консолі вводиться масив з **N** цілих чисел. Програма повинна знайти суму елементів масиву, не враховуючи всіх його найменших і найбільших елементів.

**Вхідні дані**

В першому рядку записане натуральне число **N**. У наступному рядку записані **N** цілих чисел. Всі числа за модулем не перевищують **100**.

**Вихідні дані**

Відповідь до задачі.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

7

6 2 7 1 7 1 2

**Вихідні дані #1**

10

n=**int**(input())

a=**list**(map(int,input().**split**()))

mmax=a[0]

mmin=a[0]

**for** i **in** **range**(n):

 **if** mmax>a[i]:

 mmax=a[i]

 **if** mmin<a[i]:

 mmin=a[i]

s=0

**for** i **in** **range**(n):

 **if** a[i]!=mmin and a[i]!=mmax:

 s=s+a[i]

**print**(s)

<https://www.e-olymp.com/uk/problems/4746>

Кафе "Хоботанія"

У місті Є. відкрилось нове кафе "Хоботанія", розраховане на слоників. Усі клієнти приходять у кафе в момент часу **0**, а господар вибирає у якому порядку їх обслуговувати. При цьому кожну секунду обслуговується один слоник (перший обслуговується у момент часу **0**).

Господару відомо, що якщо слоника **i** буде обслужено у момент часу **t**, то він заплатить tipsi - **t** чайових. Якщо число tipsi - **t** від'ємне, то він нічого не платить.

Допоможіть господару знайти такий порядок обслуговування слоників, який принесе йому максимальний прибуток.

**Вхідні дані**

Перший рядок містить кількість слоників **n** (**0** ≤ **n** ≤ **100**), що прийшли у кафе. Наступний рядок містить **n** чисел tips1, tips2, ..., tipsn (**0** ≤ tipsi ≤ 105).

**Вихідні дані**

Виведіть максимальний прибуток володаря кафе.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

3

3 2 3

**Вихідні дані #1**

5

x = 0

n = **int**(input())

**if** n>0:

 arr = **sorted**(list(map(int, input().**split**())), key=int, reverse=True)

 **for** i **in** **range**(n):

 **if** arr[i] - i > 0:

 x += arr[i] - i

**print**(x)

<https://www.e-olymp.com/uk/problems/8927>

Найменший дільник

Для заданого натурального числа **n** виведіть його найменший дільник, відмінний від **1**.

**Вхідні дані**

Одне натуральне число **n** (**1** < **n** < **2147000000**).

**Вихідні дані**

Виведіть найменший дільник числа **n**.

Ліміт часу **1** секунда

Ліміт використання пам'яті **128** MiB

**Вхідні дані #1**

21

**Вихідні дані #1**

3

n=int(input())

i=2

**while** n*%i!=0:*

 i=i+1

print(i)

# <https://www.e-olymp.com/uk/problems/8916>

# Перші парні

Програма має ввести з консолі натуральне число **n** та вивести в порядку зростання **n** перших парних натуральних чисел.

#### Вхідні дані:

Натуральне число **n**.

#### Вихідні дані:

В одному рядку через проміжок **n** перших парних натуральних чисел.

**Вхідні дані #1**

3

**Вихідні дані #1**

2 4 6

В одному рядку через проміжок **n** перших парних натуральних чисел

n=**int**(input())

**for** i **in** **range**(n):

 a=i\*2+2

 **print**(a,end=' ')

<https://www.e-olymp.com/uk/problems/8909>

Довжина послідовності

На вході програми маємо послідовність цілих чисел, що закінчується числом **0**. Потрібно знайти довжину даної послідовності, не враховуючи останнього нуля.

**Вхідні дані:**

Послідовність цілих чисел, по одному числу в кожному рядку.

**Вихідні дані:**

Одне число - довжина даної послідовності.

Ліміт часу **1** секунда

Ліміт використання пам'яті **64** MiB

**Вхідні дані #1**

7

-1

4

-6

0

**Вихідні дані #1**

4

k=0

while **int**(input())!=0:

 k=k+1

**print**(k)

<https://www.e-olymp.com/uk/problems/8812>

Периметр і площа

Знайдіть периметр і площу прямокутника, якщо відомі довжини його сторін.

**Вхідні дані**

Два цілі числа **a** та **b** (**a**, **b** ≤ 109) - довжини сторін прямокутника.

**Вихідні дані**

Вивести периметр та площу прямокутника.

Ліміт часу **1** секунда

Ліміт використання пам'яті **128** MiB

**Вхідні дані #1**

3 7

**Вихідні дані #1**

20 21

a,b=**map**(int, input().**split**(' '))

p=2\*(a+b)

s=a\*b

**print**(p,s)

<https://www.e-olymp.com/uk/problems/8801>

Наступне число

Прочитати ціле число та вивести наступне за ним число.

**Вхідні дані**

Одне ціле число **n**, що за модулем не більше 1018.

**Вихідні дані**

Вивести число на одиницю більше.

Ліміт часу **1** секунда

Ліміт використання пам'яті **128** MiB

**Вхідні дані #1**

10

**Вихідні дані #1**

11

a=**int**(input())

a=a+1

**print**(a)

<https://www.e-olymp.com/uk/problems/1>

<https://www.e-olymp.com/uk/problems/8810>

<https://www.e-olymp.com/uk/problems/8838>

<https://www.e-olymp.com/uk/problems/8850>

<https://www.e-olymp.com/uk/problems/8860>

<https://www.e-olymp.com/uk/problems/8862>

<https://www.e-olymp.com/uk/problems/126>

<https://www.e-olymp.com/uk/problems/8909>+

<https://www.e-olymp.com/uk/problems/8953>

<https://www.e-olymp.com/uk/problems/8954>

<https://www.e-olymp.com/uk/problems/8970>

<https://www.e-olymp.com/uk/problems/8972>

<https://www.e-olymp.com/uk/problems/4730> +

<https://www.e-olymp.com/uk/problems/8960> +

<https://github.com/maxkrivich/e-olymp/tree/master/Python>