

## Содержание

<b>03.Base [3/3]</b>	<b>3</b>
1   Задача 4А. Уставший профессор [2 sec (4 sec)]	3
2   Задача 4В. Дети знакомятся [1 sec (2 sec)]	4
3   Задача 4С. Функция [1 sec (2 sec)]	5
<b>03.Advanced [2/4]</b>	<b>6</b>
4   Задача 4D. Сумма трёх [8 sec (15 sec)]	6
5   Задача 4Е. Число [0.5 sec (1 sec)]	7
6   Задача 4F. Проверка ПСП [1 sec (2 sec)]	8
7   Задача 4G. Сумма не без разнообразия [1 sec (2 sec)]	9
<b>03.Hard [0/1]</b>	<b>10</b>
8   Задача 4H. Скобочная последовательность [2 sec (4 sec)]	10

**Общая информация:**

Вход в констест: <http://contest.yandex.ru/contest/725/>

Дедлайн на задачи 20-го октября в 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <http://compscicenter.ru/courses/algorithms-1/2014-autumn/>

Семинары ведет Сергей Владимирович Копелиович,  
контакты: [burunduk30@gmail.com](mailto:burunduk30@gmail.com), [vk.com/burunduk1](https://vk.com/burunduk1)

В каждом условии 2 таймлимита: для C/C++ и для Java, Python.

## 03.Base [3/3]

### 1 Задача 4А. Уставший профессор [2 сек (4 sec)]

Уставший профессор вечером играет в увлекательную игру.

Изначально на доске слева направо записаны целые числа  $a_1, a_2, \dots, a_n$ . Пока не уснет, профессор каждую секунду смотрит на числа, видит, что самое левое равно  $x$ , а самое правое равно  $y$ . Если  $x$  меньше, то профессор радуется, стирает слева  $x$ , а справа дописывает  $(x + y) \bmod 2^{30}$ . Иначе профессор очень расстраивается, стирает  $y$ , а слева дописывает  $(y - x) \bmod 2^{30}$ . Студенты подсчитали, что перед сном профессор успел сделать ровно  $k$  операций. Что было написано на доске, когда он наконец заснул? Для простоты можно считать, что доска в обе стороны бесконечна.

#### Формат входных данных

На первой строке  $n$  ( $1 \leq n \leq 30\,000$ ) и  $k$  ( $1 \leq k \leq 10^8$ ). На второй строке числа  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i < 10^9$ ) в порядке слева направо.

#### Формат выходных данных

На первой строке выведите все числа на доске после  $k$  операций. Выводить числа нужно в порядке слева направо.

#### Примеры

sleepgame.in	sleepgame.out
4 1 1 2 3 4	2 3 4 5
4 1000 1 2 3 4	1062488873 1072033429 1060433235 57573251
4 1 4 3 2 1	1073741821 4 3 2
4 2 4 3 2 1	5 1073741821 4 3

## 2    Задача 4В. Дети знакомятся [1 sec (2 sec)]

Мальчики и девочки со всего лагеря собрались на самой большой поляне, встали в круг, и решили познакомиться. У каждого ребенка есть имя. В определенном порядке каждый ребенок кричит «меня зовут  $X$  слева от меня стоит  $L$ , справа от меня стоит  $R$ », выходит из круга и убегает на полдник. Когда в круге остается 3 человека, они перестают кричать и спокойно расходятся. Известны имена детей и порядок, в котором они выходили из круга. Восстановите, что они кричали.

### Формат входных данных

На первой строке количество детей  $n$  ( $4 \leq n \leq 10^5$ ). На второй строке даны имена детей  $s_1, s_2, \dots, s_n$ . Дети даны в порядке против часовой стрелки и занумерованы целыми числами от 1 до  $n$ . Имена состоят из букв латинского алфавита. Длина имен не более 10. Имена могут совпадать. В третьей строке даны  $n - 3$  числа — номера детей, которые выходили из круга.

### Формат выходных данных

Для каждого вышедшего из круга ребенка выведите имена  $L$  и  $R$ , которые он прокричал.

### Примеры

meeting.in	meeting.out
9	A C
A B C D E F G H Masha	C E
2 4 7 5 3 1	F H
	C F
	A F
	Masha F

### 3    Задача 4С.    Функция [1 сек (2 сек)]

Вычислите функцию: 
$$f(n) = \begin{cases} 1 & \text{если } n \leq 2 \\ f(\lfloor 6 * n/7 \rfloor) + f(\lfloor 2 * n/3 \rfloor) & \text{если } n \bmod 2 = 1 \\ f(n-1) + f(n-3) & \text{если } n \bmod 2 = 0 \end{cases}$$

#### Формат входных данных

Входные данные содержат натуральное число  $n$  ( $1 \leq n \leq 10^{12}$ ).

#### Формат выходных данных

Выведите значение функции по модулю  $2^{32}$ .

#### Пример

function.in	function.out
7	10

## 03.Advanced [2/4]

### 4 Задача 4D. Сумма трёх [8 sec (15 sec)]

Даны три массива целых чисел  $A, B, C$  и целое число  $S$ .

Найдите такие  $i, j, k$ , что  $A_i + B_j + C_k = S$ .

#### Формат входных данных

На первой строке число  $S$  ( $1 \leq S \leq 10^9$ ). Следующие три строки содержат описание массивов  $A, B, C$  в одинаковом формате: первое число задает длину  $n$  соответствующего массива ( $1 \leq n \leq 15\,000$ ), затем заданы  $n$  целых чисел от 1 до  $10^9$  — сам массив.

#### Формат выходных данных

Если таких  $i, j, k$  не существует, выведите единственное число  $-1$ . Иначе выведите на одной строке три числа —  $i, j, k$ . Элементы массивов нумеруются с нуля. Если ответов несколько, выведите лексикографически минимальный.

#### Примеры

threesum.in	threesum.out
3 2 1 2 2 3 1 2 3 1	0 1 1
10 1 5 1 4 1 3	-1
5 4 1 2 3 4 3 5 2 1 4 5 3 2 2	0 1 2

## 5    Задача 4Е. Число [0.5 sec (1 sec)]

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрежала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

### Формат входных данных

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

### Формат выходных данных

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

### Примеры

number.in	number.out
2 20 004 66	66220004
3	3

## 6    Задача 4F. Проверка ПСП [1 sec (2 sec)]

Дана строка, состоящая из круглых, квадратных и фигурных скобок. Нужно проверить, является ли она правильной скобочной последовательностью.

### Формат входных данных

Во входном файле записана скобочная последовательность длиной не более 10 000 символов.

### Формат выходных данных

Выведите YES, если скобочная последовательность является правильной, и NO в противном случае.

### Примеры

check.in	check.out
([]())	YES
([]]	NO



## 7 Задача 4G. Сумма не без разнообразия [1 sec (2 sec)]

Задана последовательность целых чисел  $A_1, A_2, \dots, A_N$ .

Необходимо выбрать из нее подпоследовательность из подряд стоящих чисел  $A_i, A_{i+1}, \dots, A_j$  так, чтобы она содержала не менее  $K$  различных чисел, и сумма  $S = A_i + A_{i+1} + \dots + A_j$  была максимальной.

### Формат входных данных

Первая строка ввода содержит целые числа  $N$  и  $K$  ( $1 \leq K \leq N \leq 200\,000$ ).

Вторая строка содержит  $N$  целых чисел  $A_1, A_2, \dots, A_N$  ( $|A_i| \leq 1\,000\,000\,000$ ).

### Формат выходных данных

В первой строке необходимо вывести максимальное возможное значение суммы  $S$ . Во второй строке выведите индексы первого и последнего элементов найденной оптимальной подпоследовательности. Если существует несколько решений, подойдет любое из них.

Если не существует подпоследовательностей, удовлетворяющих решению задачи, выведите одну строку со словом “IMPOSSIBLE” (без кавычек).

### Примеры

threemax.in	threemax.out
7 3 -99 1 2 -100 3 2 3	-89 2 7
3 2 1 1 1	IMPOSSIBLE

## 03.Hard [0/1]

### 8    Задача 4Н. Скобочная последовательность [2 sec (4 sec)]

Дана скобочная последовательность, заданная символами «[», «]», «{», «}», «(», «)». Выведите самую длинную её подстроку, являющуюся правильной скобочной последовательностью.

**Внимание.** Решение должно работать за  $O(n)$ .

#### Формат входных данных

Входные данные содержат скобочную последовательность длины от 1 до 100 000.

#### Формат выходных данных

Выведите требуемую подстроку, возможно пустую. Если максимальных по длине правильных подстрок несколько, то выведите любую.

#### Примеры

brackets.in	brackets.out
((()()((()))	((()()((()))
]))][((())]([	[((())]
[([([([([[]]))]([([[]]([((() )[[[]]] [])))	((() )[[[]]] []
])([[]]	