

## Содержание

<b>Must have</b>	<b>2</b>
Задача 2А. Лишние пробелы [0.3 sec, 256 mb]	2
Задача 2В. Постфиксная запись [0.3 sec, 256 mb]	3
Задача 2С. Проверка ПСП [0.3 sec, 256 mb]	4
Задача 2D. Сумма простая [2.5 sec, 256 mb]	5
<b>Обязательные задачи</b>	<b>6</b>
Задача 2Е. Удалите скобки [0.3 sec, 256 mb]	6
Задача 2F. Умножение матриц [2 sec, 256 mb]	7
Задача 2G. Дети знакомятся [0.3 sec, 256 mb]	8
Задача 2H. Очередь [0.3 sec, 256 mb]	9
Задача 2I. Уставший профессор [1 sec, 256 mb]	10
<b>Дополнительные задачи</b>	<b>11</b>
Задача 2J. Baum [1 sec, 256 mb]	11
Задача 2K. Скобочная последовательность [0.2 sec, 256 mb]	12
Задача 2L. Все минимумы [0.2 sec, 256 mb]	13

---

У вас не получается читать/выводить данные?

Воспользуйтесь примерами (**c++**) (**python**).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

## Must have

### Задача 2А. Лишние пробелы [0.3 сек, 256 mb]

Дана строка. Напишите программу, которая удалит из этой строки все лишние пробелы. Пробел будем считать лишним, если:

1. он находится в самом начале строки, до самого первого слова;
2. он находится в конце строки, после самого последнего слова;
3. несколько пробелов расположены между двумя словами (проще говоря, если слова разделены более чем одним пробелом, тогда все пробелы кроме одного — лишние).

### Формат входных данных

Во входном файле записана строка, длина которой не превышает 200 символов. Строка содержит только маленькие латинские буквы и пробелы.

### Формат выходных данных

Выведите в выходной файл эту строку без лишних пробелов.

### Примеры

stdin	stdout
first test	first test

### Задача 2В. Постфиксная запись [0.3 sec, 256 mb]

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел  $A$  и  $B$  записывается как  $A B +$ . Запись  $B C + D *$  обозначает привычное нам  $(B + C) * D$ , а запись  $A B C + D * +$  означает  $A + (B + C) * D$ . Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Дано выражение в обратной польской записи. Определите его значение.

#### Формат входных данных

В единственной строке записано выражение в постфиксной записи, содержащее однозначные числа и операции  $+$ ,  $-$ ,  $*$ . Строка содержит не более 100 чисел и операций.

#### Формат выходных данных

Необходимо вывести значение записанного выражения. Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений по модулю меньше  $2^{31}$ .

#### Пример

stdin	stdout
8 9 + 1 7 - *	-102

Указание. Используйте стек.

### Задача 2С. Проверка ПСП [0.3 sec, 256 mb]

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

stdin	stdout
([] ())	YES
([])	NO

### Задача 2D. Сумма простая [2.5 sec, 256 mb]

Вам нужно научиться отвечать на запрос “сумма чисел на отрезке”.

Массив не меняется. Запросов много. Отвечать на 1 запрос следует за  $O(1)$ .

#### Формат входных данных

Размер массива —  $n$  и числа  $x, y, a_0$ , порождающие массив  $a$ :  $a_i = (x \cdot a_{i-1} + y) \bmod 2^{16}$ . Далее следуют количество запросов  $m$  и числа  $z, t, b_0$ , порождающие массив  $b$ :  $b_i = (z \cdot b_{i-1} + t) \bmod 2^{30}$ ,  $c_i = b_i \bmod n$ .  $i$ -й запрос — найти сумму на отрезке от  $\min(c_{2i}, c_{2i+1})$  до  $\max(c_{2i}, c_{2i+1})$  в массиве  $a$ .

Ограничения:  $1 \leq n \leq 10^7, 0 \leq m \leq 10^7$ . Все числа целые от 0 до  $2^{16}$ .  $t$  может быть  $-1$ .

#### Формат выходных данных

Выведите сумму всех сумм.

#### Пример

stdin	stdout
3 1 2 3	23
3 1 -1 4	

#### Замечание

$a = \{3, 5, 7\}, b = \{4, 3, 2, 1, 0, 2^{30} - 1\}, c = \{1, 0, 2, 1, 0, 0\}$ ,  
запросы =  $\{[0, 1], [1, 2], [0, 0]\}$ , суммы =  $\{8, 12, 3\}$ .

А знали ли вы, что целый 32-битный тип не «переполняется», а просто вычисляет по модулю  $2^{32}$ ? Зная это, можно все вычисления делать без модулю в типе `uint32_t`, а в конце оставить младшие 30 бит.

В Java есть знаковый сдвиг `>>` и беззнаковый `>>>`.

Одна из стандартных ошибок – переполнение типа. Проверьте, что у вас везде `int64`.

## Обязательные задачи

### Задача 2E. Удалите скобки [0.3 sec, 256 mb]

Дана строка, составленная из круглых скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

#### Формат входных данных

Во входном файле записана строка из круглых скобок. Длина строки не превосходит 100 000 символов.

#### Формат выходных данных

Выведите единственное целое число — ответ на поставленную задачу.

#### Примеры

stdin	stdout
()()	2
)()((	5
((()())	0

### Задача 2F. Умножение матриц [2 sec, 256 mb]

Даны две квадратные матрицы из целых неотрицательных чисел и целое число  $s$ . Посчитайте произведение матриц по модулю  $s$ .

#### Формат входных данных

На первой строке  $n$  ( $1 \leq n \leq 700$ ),  $s$  ( $1 \leq s \leq 10^9$ ).

Следующие  $n$  строк содержат по  $n$  целых чисел от 0 до  $s - 1$  — матрица  $A$ . Следующие  $n$  строк содержат по  $n$  целых чисел от 0 до  $s - 1$  — матрица  $B$ .

#### Формат выходных данных

Выведите  $n$  строк по  $n$  целых чисел от 0 до  $s - 1$  в каждой — матрица  $(A \times B) \bmod s$ .

#### Примеры

stdin	stdout
3 2	1 0 1
1 1 1	1 0 1
1 1 1	1 0 1
1 1 1	
1 0 0	
0 1 0	
0 1 1	

#### Замечание

Матрица  $A$  размера  $n \times m$  из целых чисел — массив `int A[n][m]`; ( $n$  строк,  $m$  столбцов).  
Произведение матриц  $A: n \times m$  и  $B: m \times k$  это матрица  $C: n \times k$

$$c_{ij} = \sum_{t=1..m} a_{it}b_{tj}$$

**Внимание:** это оптимизационная задача.

### Задача 2G. Дети знакомятся [0.3 сек, 256 mb]

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

stdin	stdout
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

#### Замечание

Есть очень простое решение без структур и указателей.

### Задача 2Н. Очередь [0.3 sec, 256 mb]

В очереди в магазин стоят люди. Человек  $i$  хочет купить товар  $a_i$ . Изначально в магазине ничего нет. Происходят события следующих типов:

1. В момент времени  $T$  поступил один экземпляр товара  $A$ .
2. В момент времени  $T$  в конец очереди встал человек, который хочет купить товар  $A$ .

Нужно промоделировать процесс и для каждого человека определить, сколько он будет стоять в очереди. Замечание: как только первый в очереди может купить то, что хочет, он сразу мгновенно покупает и уходит.

#### Формат входных данных

Число событий  $N$ ,  $1 \leq N \leq 10\,000$ . Далее события в порядке возрастания времени  $T$ . Каждое событие описывается так:  $Type\ T\ A$ , где  $Type$  — тип события.  $1 \leq T \leq 60\,000$ ,  $1 \leq A \leq 10\,000$

#### Формат выходных данных

Для каждого человека (в том порядке, в котором люди вставали в очередь) выведите, сколько человек простоял в очереди. Если он так и остался стоять, выведите -1.

#### Примеры

stdin	stdout
9	20 40 20 0 -1
2 10 1	
2 20 1	
1 30 1	
2 40 2	
1 50 2	
1 60 1	
1 70 3	
2 80 3	
2 90 1	

### Задача 21. Уставший профессор [1 sec, 256 mb]

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

stdin	stdout
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

#### Замечание

Брать по модулю  $2^{30}$  не нужно:  $(a + b) \& ((1 \ll 30) - 1)$ .

Потренируйтесь писать дек на **циклическом** массиве фиксированного размера.

## Дополнительные задачи

### Задача 2J. Baum [1 sec, 256 mb]

*Цикл* – неориентированный граф из  $n$  вершин и  $n$  рёбер, такой, что рёбра проведены между 1-й и 2-й вершинами, между 2-й и 3-й, ..., между  $(n-1)$ -й и  $n$ -й вершинами, и между  $n$ -й и 1-й.

Два игрока делят цикл между собой. Первый игрок начинает и своим ходом берёт произвольную вершину себе.

Второй игрок своим ходом может взять себе любую другую вершину.

Ходы чередуются, каждым новым ходом один из игроков берёт себе вершину, не взятую никем за предыдущие ходы.

Каждый игрок на каждом своём ходу кроме первого может взять только вершину, соединённую ребром с одной из уже выбранных им на предыдущих ходах.

Игра заканчивается, когда все вершины взяты.

У каждой вершины есть положительный целый вес. Каждый игрок хочет максимизировать сумму чисел на выбранных им вершинах.

Какая будет сумма у каждого из игроков в конце игры, если оба играют оптимально?

#### Формат входных данных

Первая строка содержит целое число  $n$  от 3 до  $10^6$ .

Вторая строка  $n$  целых чисел от 1 до  $10^9$  — веса вершин цикла в порядке обхода.

#### Формат выходных данных

Выведите два целых числа — сумма первого игрока, сумма второго игрока.

#### Примеры

stdin	stdout
4 1 2 3 4	5 5

### Задача 2К. Скобочная последовательность [0.2 sec, 256 mb]

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

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

**Определение.** Подстрока строки  $s_1s_2s_3\dots s_n$  – подотрезок  $s_is_{i+1}\dots s_j$  для некоторых  $i, j$ .

#### 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

stdin	stdout
((()((())))	((()((())))
]))][((()]](([[	][((()]]
[[([[[([[[]])]]([([[]([([()]]][[]]][])))	((()]]][[]]][]
])(([[]	

#### Замечание

Задача про стек. Решение почти не отличается от “проверки на правильность”.

### Задача 2L. Все минимумы [0.2 сек, 256 mb]

Дан массив целых чисел  $a_1, a_2, \dots, a_n$ .

Для каждого его подотрезка  $[a_L..a_R]$  определим  $F(L, R) := \min\{a_L, \dots, a_R\}$ .

Найдите

$$\sum_{1 \leq L \leq R \leq n} F(L, R)$$

то есть сумму минимумов всех подотрезков.

#### Формат входных данных

Первая строка входных данных содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — размер массива. Во второй строке через пробел заданы элементы массива, все числа целые от  $-10^6$  до  $10^6$ .

#### Формат выходных данных

Выведите единственное число — сумму минимумов всех подотрезков массива  $a$ .

#### Примеры

stdin	stdout
1 5	5
2 -10 1	-19
4 1 2 3 4	20
5 -3 2 -4 1 -5	-52

#### Замечание

Принимаются только решения за  $\mathcal{O}(n)$ . ОК-сабмиты будут подвергнуты code review.