

## Содержание

<b>Must have</b>	2
Задача 5А. Корень кубического уравнения [0.2 sec, 256 mb]	2
Задача 5В. Коровы – в стойла! [0.2 sec, 256 mb]	3
<b>Обязательные задачи</b>	4
Задача 5С. Минимальный максимум [2 sec, 256 mb]	4
Задача 5D. Поезда [0.2 sec, 256 mb]	5
Задача 5Е. Минимальное покрытие [0.2 sec, 256 mb]	6
Задача 5F. Точки и отрезки [0.2 sec, 256 mb]	7
Задача 5G. Том Сойер и его друзья [0.2 sec, 256 mb]	8
<b>Дополнительные задачи</b>	9
Задача 5H. Китайские часы [0.2 sec, 256 mb]	9
Задача 5I. Долгий путь [0.4 sec, 256 mb]	10
Задача 5J. И снова сумма... [2 sec, 256 mb]	11
Задача 5K. Инверсии отрезка [0.8 sec, 256 mb]	12

---

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 5А. Корень кубического уравнения [0.2 sec, 256 mb]

Дано кубическое уравнение  $ax^3 + bx^2 + cx + d = 0$  ( $a \neq 0$ ). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

#### Формат входных данных

Во входном файле через пробел записаны четыре целых числа:  $-1\,000 \leq a, b, c, d \leq 1\,000$ .

#### Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 6 знаков после десятичной точки.

#### Пример

stdin	stdout
1 -3 3 -1	1
-1 -6 -12 -7	-1.000000

#### Подсказка по решению

Это задача на вещественный бинпоиск. Бойтесь погрешности.  
Возможно, вам понадобится тип `long double`.

При выводе не забывайте точность вывода. Варианты для «`double x`»:

- `cout.setprecision(15); cout << x;`
- `printf("%.20f\n", x);`

### Задача 5В. Коровы – в стойла! [0.2 sec, 256 mb]

На прямой расположены стойла, в которые необходимо расставить коров так, чтобы минимальное расстояние между коровами было как можно больше.

#### Формат входных данных

В первой строке вводятся числа  $N$  ( $2 < N < 10001$ ) — количество стойл и  $K$  ( $1 < K < N$ ) — количество коров. Во второй строке задаются  $N$  целых чисел 0 до  $10^9$  в порядке возрастания — координаты стойл.

#### Формат выходных данных

Выведите одно число — наибольшее возможное допустимое расстояние.

#### Примеры

stdin	stdout
5 3 1 2 3 100 1000	99

#### Замечание

Разобрана на практике.

## Обязательные задачи

### Задача 5С. Минимальный максимум [2 сек, 256 mb]

Даны  $n$  нестрого возрастающих массивов  $A_i$  и  $m$  нестрого убывающих массивов  $B_j$ . Все массивы имеют одну и ту же длину  $l$ . Далее даны  $q$  запросов вида  $(i, j)$ , ответ на запрос – такое  $k$ , что  $\max(A_{ik}, B_{jk})$  минимален. Если таких  $k$  несколько, можно вернуть любое.

#### Формат входных данных

На первой строке числа  $n, m, l$  ( $1 \leq n, m \leq 900; 1 \leq l \leq 3000$ ). Следующие  $n$  строк содержат описания массивов  $A_i$ . Каждый массив описывается перечислением  $l$  элементов. Элементы массива – целые числа от 0 до  $10^5 - 1$ . Далее описание массивов  $B_j$  в таком же формате. Массивы и элементы внутри массива нумеруются с 1. На следующей строке число запросов  $q$  ( $1 \leq q \leq n \cdot m$ ). Следующие  $q$  строк содержат пары чисел  $i, j$  ( $1 \leq i \leq n, 1 \leq j \leq m$ ).

#### Формат выходных данных

Выведите  $q$  чисел от 1 до  $l$  – ответы на запросы.

#### Примеры

stdin	stdout
4 3 5	3
1 2 3 4 5	4
1 1 1 1 1	3
0 99999 99999 99999 99999	5
0 0 0 0 99999	4
5 4 3 2 1	3
99999 99999 99999 0 0	1
99999 99999 0 0 0	1
12	1
1 1	4
1 2	4
1 3	3
2 1	
2 2	
2 3	
3 1	
3 2	
3 3	
4 1	
4 2	
4 3	

#### Замечание

Входные данные огромны. Читайте их быстро.

### Задача 5D. Поезда [0.2 sec, 256 mb]

В связи с участвовавшим числом аварий на железнодорожной ветке Москва–Саратов, руководство железной дороги решило изменить график движения поездов. Тщательный анализ состояния железнодорожного полотна показал, что оптимальным является следующий график движения поездов с учетом остановок на станциях: сначала поезд идет на протяжении  $T_1$  минут со скоростью  $V_1$  метров в минуту, затем  $T_2$  минут со скоростью  $V_2$  метров в минуту, ..., наконец,  $T_N$  минут со скоростью  $V_N$  метров в минуту. В течение некоторых интервалов поезд может стоять (скорость равна 0).

По действующей инструкции обеспечения безопасности движения поездов расстояние между локомотивами двух следующих друг за другом поездов должно быть не менее  $L$  метров. Определите минимально допустимый интервал в минутах между отправлениями поездов, позволяющий им двигаться по этому графику без опасного сближения.

#### Формат входных данных

В первых двух строках входного файла содержится два натуральных числа, задающие минимально допустимое расстояние  $L$  и количество участков пути  $N$  ( $100 \leq L \leq 10\,000$ ,  $1 \leq N \leq 1000$ ). Далее следует  $N$  пар целых чисел  $T_i$  и  $V_i$ , задающих график движения поездов ( $1 \leq T_i \leq 1000$ ,  $0 \leq V_i \leq 1000$ ).

#### Формат выходных данных

В выходной файл необходимо вывести искомый интервал между отправлениями поездов в минутах, не менее чем с тремя верными знаками после десятичной точки.

#### Пример

stdin	stdout
1000	27.500
4	
10 0	
30 80	
15 0	
20 100	

#### Замечание

Внутри бинпоиска ещё и два указателя. Жуть.

### Задача 5Е. Минимальное покрытие [0.2 sec, 256 mb]

На прямой задано некоторое множество отрезков с целочисленными координатами концов  $[L_i, R_i]$ . Выберите среди данного множества подмножество отрезков, целиком покрывающее отрезок  $[0, M]$ , ( $M$  — натуральное число), содержащее наименьшее число отрезков.

#### Формат входных данных

В первой строке указана константа  $M$  ( $1 \leq M \leq 5\,000$ ). В каждой последующей строке записана пара чисел  $L_i$  и  $R_i$  ( $|L_i|, |R_i| \leq 50\,000$ ), задающая координаты левого и правого концов отрезков. Список завершается парой нулей. Общее число отрезков не превышает 100 000.

#### Формат выходных данных

В первой строке выходного файла выведите минимальное число отрезков, необходимое для покрытия отрезка  $[0, M]$ . Далее выведите список покрывающего подмножества, упорядоченный по возрастанию координат левых концов отрезков. Список отрезков выводится в том же формате, что и во входе. Завершающие два нуля выводить не нужно.

Если покрытие отрезка  $[0, M]$  исходным множеством отрезков  $[L_i, R_i]$  невозможно, то следует вывести единственную фразу “No solution”.

#### Примеры

stdin	stdout
1 -1 0 -5 -3 2 5 0 0	No solution
1 -1 0 0 1 0 0	1 0 1

#### Замечание

*Акция:* реши одну задачу и получи баллы и за теорию, и за практику!

### Задача 5F. Точки и отрезки [0.2 sec, 256 mb]

Дано  $n$  отрезков на числовой прямой и  $m$  точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка  $x$  считается принадлежащей отрезку с концами  $a$  и  $b$ , если выполняется двойное неравенство  $\min(a, b) \leq x \leq \max(a, b)$ .

#### Формат входных данных

Первая строка содержит два целых числа  $n$  ( $1 \leq n \leq 10^5$ ) — число отрезков и  $m$  ( $1 \leq m \leq 10^5$ ) — число точек. В следующих  $n$  строках записаны по два целых числа  $a_i$  и  $b_i$  — координаты концов соответствующего отрезка. В последней строке записаны  $m$  целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю  $10^9$ .

#### Формат выходных данных

В выходной файл выведите  $m$  чисел — для каждой точки выведите количество отрезков, в которых она содержится.

#### Примеры

stdin	stdout
2 2 0 5 7 10 1 6	1 0
1 3 -10 10 -100 100 0	0 0 1

#### Замечание

События.

### Задача 5G. Том Сойер и его друзья [0.2 сек, 256 mb]

Друзья Тома Сойера по очереди красят забор разными красками. Каждый из них красит несколько идущих подряд секций забора в определенный цвет, при этом используемые цвета могут повторяться. Новая краска ложится поверх старой. Для каждой краски вычислите количество секций, которые будут покрашены этой краской после того, как все друзья закончат работу.

#### Формат входных данных

В первой строке входного файла содержатся два целых числа:  $N$  ( $1 \leq N \leq 10^9$ ) и  $K$  ( $1 \leq K \leq 50\,000$ ) — количество секций в заборе и количество различных красок соответственно.

Во второй строке содержится единственное число  $M$  ( $0 \leq M \leq 50\,000$ ) — количество друзей Тома Сойера.

Далее следуют  $M$  строк: в  $i$ -ой строке содержится информация о работе друга, который красил забор  $i$ -ым по счету, а именно 3 целых числа  $c_i, l_i, r_i$  ( $1 \leq c_i \leq K, 1 \leq l_i \leq r_i \leq N$ ) — номер краски, которую использовал  $i$ -й друг, номер первой и номер последней покрашенной секции соответственно.

#### Формат выходных данных

Выведите в единственную строку выходного файла  $K$  целых чисел:  $i$ -ое число должно быть равно количеству секций, покрашенных  $i$ -й краской.

#### Пример

stdin	stdout
5 3 4 1 3 4 2 4 5 3 2 3 1 5 5	1 1 2
5 3 3 1 1 5 2 2 4 1 3 3	3 2 0

#### Замечание

События. Разобрана на практике.



## Дополнительные задачи

### Задача 5Н. Китайские часы [0.2 сек, 256 mb]

Когда программист Вова был в Китае, он обнаружил, что российские часы “Заря” продаются там в десять раз дешевле, чем в России. Вова решил провернуть выгодное дело и закупил огромную партию с тем, чтобы привезти часы на родину и продать их за полцены (то есть впятеро дороже, чем купил). Но дома обнаружилось, что многие часы идут вразнобой, к тому же, от малейшего толчка часы останавливаются (либо наоборот, начинают идти). Очевидно, это были не настоящие часы фирмы “Заря”, а их точные копии. Чтобы быстро продать всю партию часов, Вова хочет поставить их все на одинаковое время (тогда будет неважно, правильное это время или нет — можно будет сказать, что это местное время завода-изготовителя) и перед открыванием крышки просто встряхнуть чемодан, чтобы часы одновременно пошли.

Для установки времени на часах есть заводная головка, вращение которой позволяет быстро поворачивать стрелки, при этом часовая стрелка крутится в 60 раз медленнее минутной, а минутная — в 60 раз медленнее секундной. Один оборот заводной головки поворачивает секундную стрелку на полный оборот, и хотя на такой поворот уходит всего одна секунда, чтобы изменить время на 6 часов, потребуется 6 часов. Вращать заводную головку и стрелки можно только по часовой стрелке, чтобы не повредить хрупкий механизм часов. Помогите Вове минимизировать усилия по предпродажной подготовке часов, выбрав время, которое будет установлено на всех часах.

#### Формат входных данных

В первой строке содержится одно целое число  $n$  ( $1 \leq n \leq 50\,000$ ) — количество часов в партии. Далее следует  $n$  строк,  $(i + 1)$ -я строка ввода содержит время на  $i$ -х часах в формате  $h : mm : ss$ , где целое число  $h$  ( $1 \leq h \leq 12$ ) означает час, а двухразрядные целые числа  $mm$  и  $ss$  ( $00 \leq mm, ss \leq 59$ ) — минуты и секунды соответственно.

#### Формат выходных данных

Выведите время, которое нужно установить на всех часах, в формате, указанном выше.

#### Пример

stdin	stdout
3	12:10:01
11:30:00	
12:10:01	
6:10:18	

#### Замечание

События!

### Задача 51. Долгий путь [0.4 sec, 256 mb]

Участники китайских сборов по спортивной ходьбе идут по дороге от железнодорожной станции к месту проведения сборов. Упрощенно, дорога представляет собой прямую, причем бесконечную. В некоторый момент времени выяснилось, что все участники разбились на  $n$  групп по  $k_i$  людей в каждой. Каждая из групп в этот момент времени находится в точке с координатой  $x_i$ . Известно, что если в группе  $D$  людей, то группа движется со скоростью  $100/D$  в единицу времени (группа движется вправо — в направлении возрастания координаты). Если какая-то группа догоняет другую, то они сливаются, и соответствующим образом меняется их скорость (может слиться более двух групп одновременно). Так как дорога бесконечна, то в какой-то момент времени наступит такая ситуация, что уже ни одного слияния не произойдет. Вам требуется определить окончательное количество групп и количества людей в них.

#### Формат входных данных

В первой строке входного файла находится число  $n$  ( $1 \leq n \leq 10^5$ ). Далее следует  $n$  строк, в каждой из которых написано количество людей в соответствующей группе  $k_i$  и ее координата  $x_i$  ( $x_i$  — вещественные числа с не более чем тремя знаками после запятой и не превосходят  $10^4$  по модулю,  $1 \leq k_i \leq 100$ , все координаты различны).

#### Формат выходных данных

В первой строке выведите количество групп  $m$ , которые остались. Во второй строке выведите  $m$  чисел — количества людей в этих группах, в произвольном порядке.

#### Пример

stdin	stdout
4	2
1 0	5 5
2 9000	
4 1	
3 10000	

#### Замечание

Опять события? А как хранить?

### Задача 5J. И снова сумма... [2 sec, 256 mb]

Реализуйте структуру данных, которая поддерживает множество  $S$  целых чисел, с которым разрешается производить следующие операции:

- $add(i)$  — добавить в множество  $S$  число  $i$  (если он там уже есть, то множество не меняется);
- $sum(l, r)$  — вывести сумму всех элементов  $x$  из  $S$ , которые удовлетворяют неравенству  $l \leq x \leq r$ .

#### Формат входных данных

Исходно множество  $S$  пусто. Первая строка входного файла содержит  $n$  — количество операций ( $1 \leq n \leq 300\,000$ ). Следующие  $n$  строк содержат операции. Каждая операция имеет вид либо «+  $i$ », либо «?  $l$   $r$ ». Операция «?  $l$   $r$ » задает запрос  $sum(l, r)$ .

Если операция «+  $i$ » идет во входном файле в начале или после другой операции «+», то она задает операцию  $add(i)$ . Если же она идет после запроса «?», и результат этого запроса был  $y$ , то выполняется операция  $add((i + y) \bmod 10^9)$ .

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до  $10^9$ .

#### Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

#### Пример

stdin	stdout
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

#### Замечание

Это задача на тему «пополняемый массив».

Запрещается пользоваться деревьями и STL-контейнерами сложнее `vector`.

### Задача 5К. Инверсии отрезка [0.8 сек, 256 mb]

Дана перестановка  $p$  из  $n$  чисел. Ваша задача — быстро обрабатывать запросы вида *количество инверсий на отрезке данной перестановки*. Напомним, инверсией называется пара индексов  $(i, j)$ :  $i < j$ ,  $p_i > p_j$ .

#### Формат входных данных

На первой строке целое число  $n$ .

На второй строке сама перестановка.

На третьей строке количество запросов  $q$ .

Следующие  $q$  строк содержат пары целых чисел  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq n$ ).

#### Формат выходных данных

Для каждого из  $q$  запросов выведите ответ на отдельной строке.

**Подзадача 1**  $1 \leq n, q \leq 10^5$   $l_i \leq l_{i+1}, r_i \leq r_{i+1}$ .

**Подзадача 2**  $1 \leq n, q \leq 20\,000$   $l_i$  и  $r_i$  произвольны

#### Пример

stdin	stdout
5	3
5 4 3 1 2	9
3	2
1 3	
1 5	
3 5	

#### Замечание

К сожалению, здесь вам понадобится структура из будущего, дерево Фенвика. Если вы его пока не знаете, не беритесь писать код.