

## Задача А. Выбросить середину

Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Назовем массив  $b$  из  $k$  чисел делимым, если существует такой индекс  $1 \leq i \leq k$ , что  $\sum_{j \leq i} b_j = \sum_{q > i} b_q$ .

Дан массив чисел  $a$ , содержащий  $n$  положительных целых чисел. Вы хотите выбрать непустой подотрезок  $[l, r]$  данного массива и удалить его таким образом, чтобы оставшийся массив был бы делимым (две оставшиеся части склеиваются в один массив).

### Формат входных данных

В первой строке дано целое число  $n$  ( $3 \leq n \leq 8000$ ) — длина массива. Вторая строка содержит  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — выписанные числа. Гарантируется, что исходный массив не делим.

### Формат выходных данных

Выведите границы  $l$  и  $r$  подотрезка, который вы удаляете ( $1 \leq l \leq r \leq n$ ). Нельзя вырезать весь массив. Если решения не существует, выведите  $-1$ . Если возможных подотрезков несколько, выведите любой.

### Пример

стандартный ввод	стандартный вывод
7 7 5 9 1 3 9 6	3 4

## Задача В. Расставить ладьи

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Демид любит играть в шахматы, но часто теряет вещи и забывает правила. Родители купили ему много наборов и досок для шахмат, однако он все их потерял, кроме ладей, которые являются его любимыми фигурами. Демид решил поиграть и попытался вспомнить правила шахмат. Для начала он помнит, что существовала доска, поэтому на картоне он нарисовал доску с  $n$  рядами и  $m$  столбцами. Он решил, что каждое поле либо останется пустым, либо в него будет помещена ровно одна из шести видов игрушек: либо машинка, либо кубик с написанными на нем числами от 0 до 4.

К сожалению, единственная часть шахмат, которую Демид помнит, кроме упомянутых, это как ходит ладья. Ладья может перемещаться на любое другое поле в своем ряду или столбце за один ход. Поэтому он решил разместить определенное количество ладей на пустых полях своей доски и двигать их, не поднимая с доски. Таким образом, ладья не может пересекать поле, на котором уже находится какая-либо игрушка. Во время игры он понял, что игра не слишком интересная, если на доске слишком много ладей, так как они часто мешают друг другу, а также если на доске слишком мало ладей, так как в этом случае нечем играть. Поэтому он ввел следующие правила:

- Ни одна ладья не должна находиться на поле, на которое другая ладья может добраться за один ход.
- Если на каком-то поле лежит кубик с числом  $x$ , то на соседних полях должно находиться ровно  $x$  ладей (поля считаются соседними, если они имеют общую сторону).
- Для каждого поля, которое изначально было пустым, должна существовать ладья, которая может добраться до этого поля за максимум один ход.

Демид быстро заметил, что прежде чем начать играть, он сначала должен найти начальную расстановку ладей, если такая вообще существует, которая соблюдает указанные правила для его доски. Ваша задача — помочь Демиду начать игру, найдя заданную начальную доску и расстановку ладей, которая соблюдает все заданные правила, или определить, что такая доска не существует.

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 10$ ) — размеры доски. Следующие  $n$  строк содержат по  $m$  символов — описание доски. Точка обозначает свободное поле, цифра обозначает кубик, на котором написано соответствующее цифре число, а решетка обозначает машинку.

### Формат выходных данных

В первой строке выведите Yes или No — можно ли расставить ладей требуемым образом. Если можно, выведите дальше  $n$  строк по  $m$  символов — описание доски. Ладей обозначайте заглавной буквой R.

### Примеры

стандартный ввод	стандартный вывод
5 6 1.10#. .....2 .....0 ..... .2...#	Yes 1R10#R ...R2 .....0 ...R.. R2R...#
2 2 .0 2.	No

## Задача С. Максимальная префиксная сумма

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вам дан массив целых чисел  $a_1, a_2, \dots, a_n$ .

Определим  $f(b_1, b_2, \dots, b_k)$  как максимальную префиксную сумму массива  $b$ , то есть построим массив  $pref$  длины  $k + 1$  следующим образом:

- $pref_0 = 0$ ;
- $pref_i = pref_{i-1} + b_i$  для  $1 \leq i \leq k$ .

Тогда  $f(b_1, b_2, \dots, b_k) = \max(pref_0, pref_1, \dots, pref_k)$ .

Ваша задача найти  $\sum_{l=1}^n \sum_{r=l}^n f(a_l, a_{l+1}, \dots, a_r)$ .

### Формат входных данных

В первой строке входных данных содержится единственное целое число  $t$  ( $1 \leq t \leq 10^5$ ) — количество наборов входных данных. Далее следует их описание.

Первая строка каждого набора входных данных содержит единственное целое число  $n$  ( $1 \leq n \leq 10^5$ ) — длина массива  $a$ .

Вторая строка каждого набора входных данных содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-10^4 \leq a_i \leq 10^4$ ) — описание массива  $a$ .

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $5 \cdot 10^5$ .

### Формат выходных данных

Для каждого набора входных данных выведите  $\sum_{l=1}^n \sum_{r=l}^n f(a_l, a_{l+1}, \dots, a_r)$ .

### Пример

стандартный ввод	стандартный вывод
4	0
1	4
-1	14
2	54
2 -1	
4	
1 -3 2 2	
6	
-3 2 4 -1 0 -1	

## Задача D. Сравнение квадратных корней

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Аня загадывает два неотрицательных целых числа  $A, B$ , а Демид загадывает два неотрицательных целых числа  $C, D$ .

Необходимо узнать, что больше —  $A + \sqrt{B}$  или  $C + \sqrt{D}$ ?

### Формат входных данных

В первой строке дано число  $t$  ( $1 \leq t \leq 10000$ ) — число случаев, которые необходимо рассмотреть. В каждой из последующих  $t$  строк находится по четыре целых числа  $A_i, B_i, C_i, D_i$  ( $0 \leq A_i, B_i, C_i, D_i \leq 10^9$ ).

### Формат выходных данных

Для каждого случая выведите одну строку. Пусть  $L_i = A_i + \sqrt{B_i}$ , а  $R_i = C_i + \sqrt{D_i}$ . В строке должно содержаться слово «Less», если  $L_i < R_i$ , «Equal», если  $L_i = R_i$ , и «Greater», если  $L_i > R_i$ .

### Пример

стандартный ввод	стандартный вывод
3	Less
1 2 2 3	Equal
0 1 1 0	Greater
3 2 1 0	

## Задача Е. Художники Баянистана

Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

Давным-давно существовало королевство Баянистан. В королевстве было  $n$  городов.

Как это часто бывает, некоторые пары городов были соединены дорогами. Великий визирь Баянистана был ужасным скрягой, поэтому дорожная сеть представляла собой дерево. Дороги были сделаны из бетонных блоков, и все они имели серый цвет. Однажды великий визирь пришел к выводу, что путешествие по серым дорогам делает людей несчастными. Поскольку он не хотел рисковать революцией, он решил раскрасить дороги в счастливые цвета, семь цветов радуги.

Каждый день он выбирал два города  $s_i$  и  $t_i$  и цвет  $c_i$ . Из города  $s_i$  отправлялись художники с достаточным запасом цвета  $c_i$ , в город  $t_i$  по кратчайшему пути. По пути, всякий раз, когда они встречали дорогу, не имеющую цвета  $c_i$ , они красили ее в этот цвет. Для покраски одной дороги использовался один литр краски.

Для каждого цвета вычислите сколько литров краски этого цвета суммарно было затрачено.

### Формат входных данных

Первая строка содержит положительное целое число  $n$  ( $2 \leq n \leq 10^6$ ). Для удобства города пронумерованы от 0 до  $n - 1$ . Далее следует  $n - 1$  строка. Каждая из этих строк содержит два числа  $a_i$  и  $b_i$  ( $0 \leq a_i, b_i < n$ ) — конечные точки одной дороги.

Следующая строка содержит одно целое число  $q$  ( $1 \leq q \leq 2 \cdot 10^5$ ), указывающее количество приказов.

Далее следует  $q$  строк, каждая из которых описывает один приказ. Каждое описание состоит из двух целых чисел  $s_i$  и  $t_i$  ( $0 \leq s_i, t_i < n$ ) и строки  $c_i$  ( $c_i$  — один из следующих цветов: red, orange, yellow, green, blue, indigo, violet).

### Формат выходных данных

Выведите семь строк. Каждая из этих строк должна содержать название цвета (в указанном выше порядке) и количество литров краски, которое будет использовано.

## Примеры

стандартный ввод	стандартный вывод
8 0 4 7 5 5 6 3 5 1 4 4 2 4 3 4 0 7 red 2 6 orange 4 3 green 1 7 green	red 4 orange 4 yellow 0 green 4 blue 0 indigo 0 violet 0
12 0 1 1 2 2 3 2 4 2 5 0 6 6 7 7 8 7 9 7 10 11 6 3 3 8 blue 6 1 violet 5 11 blue	red 0 orange 0 yellow 0 green 0 blue 10 indigo 0 violet 2

## Задача F. Блинчиковая Физтеха

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Самым популярным блюдом в меню блинчиковой на Физтехе является блинная башня, представляющая собой стопку из  $n$  блинов. Ресторан готовит блины трех разных вкусов, которые мы назовем  $A$ ,  $B$  и  $C$ .

Назовем башню блинов хорошей, если расположение блинов удовлетворяет следующим условиям:

- Во всех парах блинов со вкусом  $A$  и блинов со вкусом  $B$  блин со вкусом  $A$  находится выше блина со вкусом  $B$ .
- Во всех парах блинов со вкусом  $A$  и блинов со вкусом  $C$  блин со вкусом  $A$  находится выше блина со вкусом  $C$ .
- Во всех парах блинов со вкусом  $B$  и блинов со вкусом  $C$  блин со вкусом  $B$  находится выше блина со вкусом  $C$ .

Например, блинная башня с блинами со вкусами AABVBC, ACC и BBBB сверху вниз является хорошей блинной башней, а блинная башня с блинами со вкусами AABAVCC и CA не является хорошей блинной башней.

Вы можете за одну операцию вставить лопатку после  $k$ -го сверху блина в одной стопке и перевернуть стопку из верхних  $k$  блинов. Для каждой операции вы можете выбрать любое  $k$ . Перед вами  $q$  тарелок, на каждой из которых лежит блинная башня. Для каждой тарелки определите, за какое наименьшее число операций можно сделать стопку блинов хорошей.

### Формат входных данных

Первая строка содержит два числа  $n$  и  $q$  ( $2 \leq n \leq 13$ ,  $1 \leq q \leq 10^5$ ). Следующие  $q$  строк содержат по  $n$  символов  $A, B, C$  — описания башен в порядке блинов сверху вниз.

### Формат выходных данных

Для каждой башни выведите минимальное требуемое число операций, чтобы сделать башню блинов хорошей.

### Пример

стандартный ввод	стандартный вывод
5 3	0
AAAAA	2
ССВАВ	3
АВСВА	

## Задача G. Очередь за подарками

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Студенты стоят в очереди за подарками. У дарящего есть список студентов, которые должны получить подарки. Студенты из очереди могут встречаться в списке больше одного раза, а могут ни разу не встречаться. В списке не может встретиться студент, которого нет в очереди.

Студенты по очереди подходят к дарящему, и если его фамилия совпадает с верхней невычеркнутой фамилией в списке, то он получает подарок, а его фамилия вычеркивается из списка. Вне зависимости от того, получил он подарок или нет, студент возвращается в очередь и может встать в любое место этой очереди, необязательно в конец. Назовем одну итерацию этого процесса проверкой.

Зная содержимое списка и позицию каждого студента в начальной очереди, какое минимальное количество проверок может потребоваться для того, чтобы каждое имя было вычеркнуто из списка, если студенты оптимально выберут свои новые позиции в очереди?

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 3 \cdot 10^5$ ) – общее количество студентов в очереди и количество имен в списке соответственно.

Вторая строка содержит  $m$  целых чисел  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq n$ ) – имена студентов, как они появляются в списке (сверху вниз).

Третья строка содержит  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq n$ ) – начальный порядок студентов по мере их появления в очереди, где  $b_i$  – номер  $i$ -го студента в очереди. Студенты пронумерованы от 1 до  $n$ .

### Формат выходных данных

Вывод состоит из двух строк:

- Первая строка вывода это  $k$  – минимальное количество проверок, которое необходимо провести дарящему.
- Вторая строка вывода состоит из  $k$  целых чисел. Для каждой проверки выводится число, обозначающее, на какую позицию в очереди студент должен вернуться после проверки.

Если существует несколько решений, которые приводят к минимальному количеству проверок, то выведите любое из них.

### Пример

стандартный ввод	стандартный вывод
4 5	7
4 1 2 4 4	4 4 2 4 4 1 4
4 3 2 1	



## Задача Н. Сгенерированная последовательность

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Есть четыре целых числа  $n, a, b$  и  $p$ , таких, что  $n$  и  $a$  взаимнопростые. Компьютер генерирует последовательность  $c_0, c_1, \dots, c_{n-1}$ , где  $c_i = 0$ , если  $(ai + b) \bmod n < p$ , иначе  $c_i = 1$ . У вас есть другая, более короткая последовательность из  $m$  элементов  $w_0, w_1, \dots, w_{m-1}$ . Задача состоит в том, чтобы как можно быстрее найти число вхождений более короткой последовательности в последовательность, сгенерированную компьютером.

### Формат входных данных

В первой строке даны пять целых чисел  $n, a, b, p$  и  $m$  ( $2 \leq n \leq 10^9, 1 \leq p, a, b, m < n, 1 \leq m \leq 10^6$ ). Числа  $n$  и  $a$  взаимнопростые. Вторая строка содержит последовательность  $w$  длины  $m$  из нулей и единиц.

### Формат выходных данных

Выведите одно число — ответ на задачу.

### Пример

стандартный ввод	стандартный вывод
9 5 6 4 3 101	3

### Замечание

$i$	0	1	2	3	4	5	6	7	8
$ai + b$	6	11	16	21	26	31	36	41	46
$(ai + b) \bmod n$	6	2	7	3	8	4	0	5	1
$c_i$	1	0	1	0	1	1	0	1	0

## Задача I. Простые пятизначные числа

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Загаданы три простых пятизначных числа. Вам даны пять чисел  $a_i$ , равные сумме цифр загаданных чисел в разряде  $i$  (пронумерованы от младшего к старшему). Восстановите загаданные числа.

### Формат входных данных

В первой строке дано целое число  $T$  ( $1 \leq T \leq 100$ ) — количество тестовых случаев. Далее идут  $T$  строк, каждая из которых содержит пять целых чисел  $a_i$  ( $0 \leq a_i \leq 30$ ).

### Формат выходных данных

Для каждого тестового случая выведите три простых загаданных пятизначных числа в любом порядке. Если ответов несколько, выведите любой. Если ответа на тестовый случай не существует, выведите  $-1$ .

### Пример

стандартный ввод	стандартный вывод
2	20533 87119 44987
19 12 15 11 14	-1
22 19 3 8 23	

## Задача J. Онлайн-кинотеатр

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вы смотрите одно ТВ-шоу в двух разных онлайн-кинотеатрах. Когда в кинотеатре выходит эпизод, у вас должна быть подписка этого кинотеатра, чтобы посмотреть её. Подписку можно купить в любой день, и она будет действовать  $k$  дней. В течение каждого из этих дней вы можете смотреть эпизоды в кинотеатре, на который вы только что купили подписку.

Учитывая даты выхода новых эпизодов в каждом онлайн-кинотеатре в течение следующих  $n$  дней, определите общее количество периодов подписки, которые вам нужно купить, чтобы каждый день иметь возможность смотреть все серии. Если эпизод выходит в обоих кинотеатрах одновременно, вы можете посмотреть его в любом из них.

### Формат входных данных

Первая строка содержит  $n$  ( $1 \leq n \leq 4 \cdot 10^5$ ) — количество дней, на которые у вас есть расписание выхода новых серий, и  $k$  ( $1 \leq k \leq n$ ) — количество дней, на которые рассчитана каждая подписка. Следующая строка содержит строку из  $n$  символов, состоящую из символов 1, 2 или B.

Символ  $i$  описывает, выходит ли новый эпизод в первом кинотеатре (1), втором кинотеатре (2) или в обоих кинотеатрах (B) в день  $i$ .

### Формат выходных данных

Выведите одно число — ответ на задачу.

### Пример

стандартный ввод	стандартный вывод
8 3 1B21B212	5

## Задача К. Черно-белый граф

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Есть неориентированный граф из  $n$  вершин и  $m$  ребер. Вы должны раскрасить ребра в два цвета так, чтобы для каждой вершины со степенью хотя бы два было выполнено, что из нее выходит хотя бы одно ребро каждого из цветов. Другими словами, каждой вершине со степенью хотя бы два инцидентно хотя бы одно ребро каждого из цветов.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ). Следующие  $m$  строк содержат по два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i \neq v_i \leq n$ ) — номера вершин, соединяемых ребром. В графе нет кратных ребер.

### Формат выходных данных

Если нельзя раскрасить граф требуемым образом, выведите *No*. Иначе в первой строке выведите *Yes*, а во второй выведите  $m$  чисел 1 и 2 — цвета ребер. Черные ребра обозначайте числом 1, а белые числом 2.

### Примеры

стандартный ввод	стандартный вывод
5 6 2 1 3 2 3 1 3 4 1 4 4 5	Yes 1 2 1 2 2 1
7 7 2 1 3 2 1 3 4 5 5 6 6 7 7 4	No

## Задача L. Посадка в самолет

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Представим самолет как числовую прямую. В самолете  $n$  рядов, будем считать, что  $i$ -й ряд имеет координату  $x = i$ . В самолет заходят  $n$  пассажиров,  $i$ -й пассажир хочет сесть на ряд  $r_i$ , по одному пассажиру на ряд. При входе в самолет пассажиры выстроились в очередь, можно считать, что в начале пассажир  $i$  имеет координату  $i - n$ .

Каждую секунду каждый пассажир делает шаг вправо (в сторону увеличения координаты  $x$ ), если это место свободно или пассажир справа тоже делает шаг. Когда пассажир  $i$  достигает ряда  $r_i$ , он тратит  $t_i$  секунд, чтобы раздеться и положить багаж, а затем садится на место и больше никому не мешает. Пока пассажир кладет багаж, координата его ряда заблокирована, и пассажир слева не может сделать шаг в эту координату.

Определите, сколько секунд потребуется, чтобы все пассажиры сели на свои места.

### Формат входных данных

В первой строке дано целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ). Следующие  $n$  строк содержат по два целых числа  $r_i$  и  $t_i$  ( $1 \leq r_i \leq n$ ,  $1 \leq t_i \leq 10^9$ ) — характеристики  $i$ -го пассажира. Гарантируется, что все  $r_i$  уникальны.

### Формат выходных данных

Выведите одно число — ответ на задачу.

### Пример

стандартный ввод	стандартный вывод
3 2 5 3 10 1 5	19

### Замечание

Расположение пассажиров в примере

$t \backslash x$	-2	-1	0	1	2	3
0	1	2	3	.	.	.
1	.	1	2	3	.	.
6	.	1	2	.	.	.
7	.	.	1	2	.	.
8	.	.	.	1	2	.
9	.	.	.	.	1	2
14	.	.	.	.	.	2
19	.	.	.	.	.	.

## Задача М. Life on Mars?

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В марсианском часе  $M$  марсианских минут. Марсианский час и марсианская минута не обязательно совпадают с земными, а именно, земной час в точности равен  $C$  марсианским минутам.

Алиса подарила Бобу земные часы с кукушкой. Каждые  $C$  марсианских минут они громко отбивают начало земного часа. Когда это произошло в первый раз, Боб подскочил от неожиданности и непроизвольно посмотрел на свои марсианские часы; минутная стрелка на них показывала  $A$  минут.

После  $K$ -го боя часов Боб задумался над тем, сколько минут в среднем показывают его марсианские часы в начале земного часа. Он догадывается, что среднее арифметическое этих значений со временем стремится к  $A \bmod \gcd(M, C) + \frac{M - \gcd(M, C)}{2}$ , поэтому вместо этого он решил рассмотреть их среднее квадратичное. Помогите ему, вычислив сумму квадратов значений минутной стрелки по всем  $K$  моментам, когда били часы.

Поскольку ответ может не поместиться в 64-битный целочисленный тип, выведите его по модулю  $10^9 + 7$ .

Более формально, найдите:

$$\left( \sum_{n=0}^{K-1} ((Cn + A) \bmod M)^2 \right) \bmod (10^9 + 7)$$

### Формат входных данных

Первая строка ввода содержит одно целое число  $T$  ( $1 \leq T \leq 100$ ) — количество наборов входных данных.

Единственная строка каждого набора входных данных содержит четыре целых числа  $C$ ,  $A$ ,  $M$  и  $K$  ( $1 \leq C \leq 10^9$ ,  $0 \leq A < M \leq 10^9$ ,  $1 \leq K \leq 10^9$ ).

### Формат выходных данных

Для каждого набора входных данных выведите ответ на задачу — значение выражения выше.

### Пример

стандартный ввод	стандартный вывод
2	0
1 0 1 5	25
3 4 7 3	