


[Kap6502](#) | [Выйти](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

E. Ива и Пав

ограничение по времени на тест: 5 секунд
ограничение по памяти на тест: 256 мегабайт

Ива дала Паву массив a из n элементов.

Пусть $f(l, r) = a_l \& a_{l+1} \& \dots \& a_r$ (здесь $\&$ обозначает побитовое И).

Обратите внимание, что $f(l, r)$ не определено, если $l > r$.

Ива также дала Паву q запросов.

Каждый запрос состоит из 2 чисел, k и l , и она хочет, чтобы Пав нашел наибольший индекс r ($l \leq r \leq n$), такой что $f(l, r) \geq k$.

Пав хочет решить эту задачу быстро, потому что он не хочет расстроить Иву. Он нуждается в вашей помощи.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина массива a .

Вторая строка каждого набора содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — элементы массива a .

Третья строка каждого набора содержит одно целое число q ($1 \leq q \leq 10^5$) — количество запросов, которые Ива дала Паву.

Следующие q строк каждого набора содержат по два числа, l и k ($1 \leq l \leq n$, $1 \leq k \leq 10^9$) — левая граница для отрезка и целое число k , описанное в условии.

Гарантируется, что сумма n по всем наборам не превышает $2 \cdot 10^5$. Также гарантируется, что сумма q по всем наборам не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого запроса выведите максимальный индекс r ($l \leq r \leq n$) такой, что $a_l \& a_{l+1} \& \dots \& a_r \geq k$.

Если такого r не существует, выведите -1 .

Пример

входные данные	Скопировать
3	
5	
15 14 17 42 34	
3	
1 7	
2 15	
4 5	
5	
7 5 3 1 7	
4	
1 7	
5 7	
2 3	
2 2	
7	
19 20 15 12 21 7 11	
4	
1 15	
4 4	
7 12	
5 7	
выходные данные	Скопировать
2 -1 5	
1 5 2 2	
2 6 -1 5	

Примечание

В первом наборе примера $n = 5$, и массив $a = [15, 14, 17, 42, 34]$

Первый запрос просит найти наибольший индекс r , такой что $f(1, r) \geq 7$.

$f(1, 1) = 15, f(1, 2) = 14, f(1, 3) = 0, f(1, 4) = 0, f(1, 5) = 0$, поэтому ответ $r = 2$.

Второй запрос просит найти $f(2, r) \geq 15$. Поскольку такой r не существует, ответ -1 .

Третий запрос просит найти $f(4, r) \geq 5$. $f(4, 4) = 42, f(4, 5) = 34$, поэтому ответ $r = 5$.

Во **втором наборе примера** $n = 5$, и массив $a = [7, 5, 3, 1, 7]$.

Для первого запроса, $f(1, r) \geq 7$.

$f(1, 1) = 7, f(1, 2) = 5, f(1, 3) = 1, f(1, 4) = 1, f(1, 5) = 1$, поэтому ответ на этот запрос 1 .

Для второго запроса, $f(5, r) \geq 7$.

$f(5, 5) = 7$, поэтому ответ 5 .

Для третьего запроса, $f(2, r) \geq 3$.

$f(2, 2) = 5, f(2, 3) = 1, f(2, 4) = 1, f(2, 5) = 1$, поэтому ответ 2 .

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:22:56_{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



| ITMO


[Kap6502](#) | [Выход](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Прибавления в соседей

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Поликарпу подарили массив $a[1 \dots n]$ из n целых чисел. Он может производить с массивом a следующую операцию не более n раз:

- Поликарп выбирает индекс i и прибавляет в **одного на свой выбор** из его соседей значение a_i . Более формально, Поликарп прибавляет значение a_i либо к a_{i-1} , либо к a_{i+1} (если такого соседа не существует, то и прибавить в него нельзя);
- После прибавления Поликарп удаляет i -й элемент из массива a (заметим, что в процессе удаления длина массива a уменьшается на 1).

Два пункта выше совместно обозначают одну операцию.

Например, если у Поликарпа есть массив $a = [3, 1, 6, 6, 2]$, то он может произвести с ним следующую последовательность операций:

- Поликарп выбирает $i = 2$ и прибавляет значение a_i к $(i-1)$ -му элементу: $a = [4, 6, 6, 2]$;
- Поликарп выбирает $i = 1$ и прибавляет значение a_i к $(i+1)$ -му элементу: $a = [10, 6, 2]$;
- Поликарп выбирает $i = 3$ и прибавляет значение a_i к $(i-1)$ -му элементу: $a = [10, 8]$;
- Поликарп выбирает $i = 2$ и прибавляет значение a_i к $(i-1)$ -му элементу: $a = [18]$;

Заметьте, что Поликарп мог прекратить производить операции в любой из моментов.

Поликарпу стало интересно, сколько минимум операций ему надо произвести, чтобы все элементы массива a стали одинаковыми (то есть, он хочет, чтобы все a_i были равны между собой).

Входные данные

В первой строке находится одно целое число t ($1 \leq t \leq 3000$) — количество наборов входных данных. Далее следуют t наборов входных данных.

В первой строке каждого набора содержится одно целое число n ($1 \leq n \leq 3000$) — длина массива. В следующей строке содержатся n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^5$) — массив a .

Гарантируется, что сумма n по всем наборам входных данных не превосходит 3000.

Выходные данные

Для каждого набора входных данных выведите одно число — минимальное количество операций, которые нужно сделать Поликарпу, чтобы все элементы массива a оказались одинаковыми.

Пример

входные данные	Скопировать
4 5 3 1 6 6 2 4 1 2 2 1 3 2 2 2 4 6 3 2 1	
выходные данные	Скопировать
4 2 0 2	

Примечание

В первом наборе входных данных ответ может быть получен следующим образом (один из возможных вариантов):

$[3, 1, 6, 6, 2] \xrightarrow{i=4, \text{ прибавляет влево}} [3, 1, 12, 2] \xrightarrow{i=2, \text{ прибавляет вправо}} [3, 13, 2] \xrightarrow{i=1, \text{ прибавляет вправо}} [16, 2] \xrightarrow{i=2, \text{ прибавляет влево}} [18]$.
Все элементы в массиве [18] одинаковые.

Во втором наборе входных данных ответ может быть получен следующим образом (один из возможных вариантов):

$[1, 2, 2, 1] \xrightarrow{i=1, \text{ прибавляет вправо}} [3, 2, 1] \xrightarrow{i=3, \text{ прибавляет влево}} [3, 3]$. Все элементы в массиве $[3, 3]$ одинаковые.

В третьем наборе входных данных Поликарпу не надо производить ни одну операцию, так как массив $[2, 2, 2]$ уже содержит одинаковые элементы.

В четвертом наборе входных данных ответ может быть получен следующим образом (один из возможных вариантов):

$[6, 3, 2, 1] \xrightarrow{i=3, \text{ прибавляет вправо}} [6, 3, 3] \xrightarrow{i=3, \text{ прибавляет влево}} [6, 6]$. Все элементы в массиве $[6, 6]$ одинаковые.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:22:57^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ITMO


[Kap6502](#) | [Выход](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Продуктивная встреча

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На встрече собрались n человек. В каждый момент времени любые два человека могут отойти и поговорить наедине. Одни и те же два человека могут поговорить несколько (сколько угодно) раз за встречу.

У каждого человека есть ограниченная общительность. Общительность i -го человека равна неотрицательному целому числу a_i . Это означает, что после a_i личных разговоров этот человек уходит со встречи (и больше ни с кем не разговаривает). Если $a_i = 0$, i -й человек уходит со встречи сразу после ее начала.

Встреча считается тем более *продуктивной*, чем больше в течение нее состоялось разговоров.

По данному массиву общительности a определите, какие люди должны разговаривать друг с другом, чтобы суммарное количество разговоров было как можно больше.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

В следующих $2t$ строках даны описания наборов входных данных.

В описании каждого набора данных первая строка содержит целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество людей на встрече, а во второй строке через пробел перечислены n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 2 \cdot 10^5$) — параметры общительности всех людей.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$. Также гарантируется, что сумма всех a_i (по всем наборам входных данных и всем i) не превосходит $2 \cdot 10^5$.

Выходные данные

Выполните t ответов на все наборы входных данных.

В первой строке ответа выведите число k — максимальное количество разговоров, которое возможно провести за встречу.

В каждой из следующих k строк выведите через пробел по два целых числа i и j ($1 \leq i, j \leq n$ и $i \neq j$) — номера людей, между которыми состоится очередной разговор.

Если ответов несколько, выведите любой из них.

Пример

входные данные	Скопировать
8	
2	
2 3	
3	
1 2 3	
4	
1 2 3 4	
3	
0 0 2	
2	
6 2	
3	
0 0 2	
5	
8 2 0 1 1	
5	
0 1 0 0 6	
выходные данные	Скопировать
2	
1 2	
1 2	
3	
1 3	
2 3	
2 3	
5	
1 3	
2 4	

```
2 4
3 4
3 4
0
2
1 2
1 2
0
4
1 2
1 5
1 4
1 2
1
5 2
```

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:22:59^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ITMO


[Kap6502](#) | [Выйти](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Найти В

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Массив a длины m назовем красивым, если существует массив b из m целых чисел, для которого выполняются следующие условия:

1. $\sum_{i=1}^m a_i = \sum_{i=1}^m b_i;$
2. $a_i \neq b_i$ для каждого индекса i от 1 до m ;
3. $b_i > 0$ для каждого индекса i от 1 до m .

Дан массив c длины n . Каждый элемент этого массива больше 0.

Вам нужно ответить на q запросов. Во время i -го запроса вам нужно определить, является ли подмассив $c_{l_i}, c_{l_i+1}, \dots, c_{r_i}$ красивым.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Первая строка каждого набора содержит два целых числа n и q ($1 \leq n, q \leq 3 \cdot 10^5$) — длину массива c и количество запросов.

Вторая строка каждого набора содержит n целых чисел c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^9$).

Затем следуют q строк. i -я из них содержит два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) — границы i -го подмассива.

Дополнительные ограничения на входные данные: сумма n по всем наборам входных данных не превосходит $3 \cdot 10^5$; сумма q по всем наборам входных данных не превосходит $3 \cdot 10^5$.

Выходные данные

Для каждого запроса выведите YES, если подмассив красивый. В противном случае выведите NO.

Вы можете выводить каждую букву в любом регистре (как строчную или как заглавную). Например, строки yEs, yes, Yes и YES будут приняты как положительный ответ.

Пример

входные данные	Скопировать
1 5 4 1 2 1 4 5 1 5 4 4 3 4 1 3	
выходные данные	Скопировать
YES NO YES NO	



ITMO

[Kap6502](#) | [Выйти](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Смайло и монстры

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Мальчик Смайло играет в новую игру! В игре есть n орд монстров, в i -й орде a_i монстр. Цель игры — уничтожить всех монстров. Для этого у вас есть два типа атак и счетчик комбо x , изначально равный 0:

- Первый тип атаки: вы выбираете число i от 1 до n такое, что в орде с номером i остался хотя бы один монстр. Затем вы убиваете одного монстра из орды с номером i , и счетчик комбо x увеличивается на 1.
- Второй тип атаки: вы выбираете число i от 1 до n такое, что в орде с номером i осталось хотя бы x монстров. Затем вы применяете ультимейт и убиваете x монстров из орды под номером i . После этого x обнуляется.

Ваша задача — зачистить всех монстров, то есть сделать так, чтобы ни в одной орде не осталось ни одного монстра. Смайло хочет как можно скорее победить, поэтому скажите, какое минимальное число атак можно применить?

Входные данные

В первой строке содержится единственное число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — число орд монстров.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — количества монстров в каждой орде.

Гарантируется, что сумма n по всем тестовым наборам не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите минимальное число атак, которое нужно, чтобы убить всех монстров.

Пример

входные данные	Скопировать
4 4 1 3 1 1 4 1 2 1 1 6 3 2 1 5 2 4 2 1 6	
выходные данные	Скопировать
4 4 11 5	

Примечание

В первом наборе входных данных можно применить атаку первого типа по одному разу на 1-ю, 3-ю и 4-ю орду, затем добить атакой второго типа 2-ю орду. Итого нужно 4 атаки.

Во втором наборе входных данных можно применить атаку первого типа по одному разу на 1-ю, 3-ю орду, затем добить атакой второго типа 2-ю орду, затем применить атаку первого типа один раз на 4-ю орду. Итого нужно 4 атаки.

В четвёртом наборе входных данных можно применить атаку первого типа один раз на 1-ю орду, два раза на 2-ю орду, затем применить один раз атаку второго типа на 2-ю орду, затем применить один раз атаку первого типа на 2-ю орду. Итого нужно 5 атак.

[Kap6502](#) | [Выйти](#)

ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

D. Лунный новый год и прогулка

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Приближается лунный новый год, а Боб решил отправиться на прогулку в ближайший парк.

Парк может быть представлен как связный граф из n вершин и m неориентированных ребер. Изначально Боб находился в вершине 1 и записал 1 в свою записную книжку. Он может переходить из одной вершины в другую по данным ребрам. Каждый раз, когда он посещает вершину, еще не записанную в его книжку, он записывает ее. После того, как он посетит все вершины как минимум по разу, он закончит прогулку, а в его книжке будет записана перестановка вершин a_1, a_2, \dots, a_n .

Гулять скучно, а решать задачи — интересно. Боб хочет узнать лексикографически минимальную перестановку, которую он может получить по итогам прогулки. Бобу эта задача кажется простой, поэтому он отдал ее вам.

Последовательность x лексикографически меньше последовательности y , если и только если выполняется один из следующих пунктов:

- x — префикс y , но $x \neq y$ (обратите внимание, в этой задаче такое невозможно, так как все рассматриваемые последовательности имеют одинаковую длину);
- в первой позиции, где x и y различны, в последовательности x элемент меньше, чем соответствующий элемент в y .

Входные данные

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 10^5$) — числе вершин и ребер в графе, соответственно.

Следующие m строк описывают неориентированные ребра графа. i -я из этих строк содержит два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$) — вершины, соединенные i -м ребром.

Обратите внимание, что в графе могут быть кратные ребра и петли. Гарантируется, что граф связный.

Выходные данные

Выведите лексикографически наименьшую последовательность a_1, a_2, \dots, a_n из тех, которые может записать Боб.

Примеры

входные данные	Скопировать
3 2 1 2 1 3	
выходные данные	Скопировать
1 2 3	
входные данные	Скопировать
5 5 1 4 3 4 5 4 3 2 1 5	
выходные данные	Скопировать
1 4 3 2 5	
входные данные	Скопировать
10 10 1 4 6 8 2 5 3 7 9 4 5 6 3 4 8 10 8 9 1 10	
выходные данные	Скопировать

1 4 3 7 9 8 6 5 2 10

Примечание

В первом примере одним из возможных путей Боба является путь $1 \rightarrow 2 \rightarrow 1 \rightarrow 3$. Боб в этом случае запишет последовательность $\{1, 2, 3\}$, которая является лексикографически наименьшей.

Во втором примере Боб может пойти по пути $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 5$. Тогда он запишет последовательность $\{1, 4, 3, 2, 5\}$, которая является лексикографически наименьшей.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 03.11.2025 12:23:38^{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ITMO


[Kap6502](#) | [Выходи](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Кондиционер

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Гильдонг — владелец ресторана Пулькоги. Ресторан пользуется большим спросом, и часто посетители хотят забронировать столик до посещения.

Гильдонг так старается угодить всем посетителям, что он даже запомнил предпочтаемые диапазоны температуры всех посетителей! Сматывая на список бронирований, он хочет удовлетворить всех посетителей, контролируя температуру ресторана.

В ресторане стоит кондиционер, у которого есть три состояния: *выключенное*, *нагрев*, и *охлаждение*. Когда кондиционер *выключен*, температура в ресторане не изменяется. Когда включен *нагрев*, температура увеличивается на 1 каждую минуту. Наконец, когда включено *охлаждение*, температура уменьшается на 1 каждую минуту. Гильдонг может переключать состояние сколько угодно раз, в любые целочисленные минуты. Кондиционер исходно *выключен*.

Каждый посетитель характеризуется тремя значениями: t_i — момент визита (в минутах) i -го посетителя, l_i — нижняя граница предпочтаемого диапазона температур, and h_i — верхняя граница предпочтаемого диапазона температур.

Каждый посетитель удовлетворен, если температура находится в их предпочтительном диапазоне в момент их посещения ресторана. Формально, i -й посетитель удовлетворён, если и только если в минуту t_i температура находится в отрезке от l_i до h_i , включительно.

Вам дана исходная температура, список времен посещения посетителей и их предпочтаемые диапазоны температуры, помогите ему удовлетворить всех посетителей.

Входные данные

Каждый тест состоит из одного или большего числа наборов входных данных. В первой строке записано количество наборов входных данных q ($1 \leq q \leq 500$). Затем следуют их описания.

В первой строке каждого набора входных данных записаны два целых числа n и m ($1 \leq n \leq 100$, $-10^9 \leq m \leq 10^9$), где n это число посетителей и m это изначальная температура ресторана.

Затем следуют n строк. i -я из них содержит три целых числа t_i , l_i , и h_i ($1 \leq t_i \leq 10^9$, $-10^9 \leq l_i \leq h_i \leq 10^9$), где t_i это момент визита i -го посетителя, l_i это нижняя граница предпочтаемого диапазона температур, а h_i это верхняя граница предпочтаемого диапазона температур. Все границы диапазонов — **включительны**.

Посетители даны в порядке неубывания момента посещения, а текущее время — 0.

Выходные данные

Для каждого набора входных данных выведите «YES» если возможно удовлетворить потребности всех посетителей. В противном случае, выведите «NO».

Вы можете выводить все буквы в любом регистре (верхнем или нижнем).

Пример

входные данные	Скопировать
<pre>4 3 0 5 1 2 7 3 5 10 -1 0 2 12 5 7 10 10 16 20 3 -100 100 0 0 100 -50 50 200 100 100 1 100 99 -100 0</pre>	
выходные данные	Скопировать
<pre>YES NO YES NO</pre>	

Примечание

В первом наборе входных данных примера Гильдонг может контролировать кондиционер, чтобы удовлетворить всех посетителей, следующим образом:

- В 0-ю минуту, поменяйте состояние на *нагрев* (температура 0).
- Во 2-ю минуту, поменяйте состояние на *выключенное* (температура 2).
- В 5-ю минуту, поменяйте состояние на *нагрев* (температура 2, 1-й посетитель уже удовлетворен).
- В 6-ю минуту, поменяйте состояние на *выключенное* (температура 3).
- В 7-ю минуту, поменяйте состояние на *охлаждение* (температура 3, 2-й посетитель удовлетворен).
- В 10-ю минуту, температура будет 0, что удовлетворит последнего посетителя.

В третьем наборе входных данных примера поменяйте состояние на *выключенное* в 0-ю минуту и оставьте его таким. Тогда все посетители будут удовлетворены. Обратите внимание, что момент посещения 1-го посетителя совпадает с моментом посещения 2-го посетителя.

Во втором и четвертом наборах входных данных примера Гильдонгу придется оставить хотя бы одного посетителя неудовлетворенным.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:23:40^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

[Kap6502](#) | [Выходи](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Максимальная ширина

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

У вашего одноклассника, которого вы не очень любите за его занудство, но уважаете за его ум, были обнаружены две строки: строка s длины n и строка t длины m .

Последовательность индексов p_1, p_2, \dots, p_m , где $1 \leq p_1 < p_2 < \dots < p_m \leq n$, называется *хорошой*, если $s_{p_i} = t_i$ для всех i от 1 до m . Шириной последовательности называется величина $\max_{1 \leq i < m} (p_{i+1} - p_i)$.

Помогите однокласснику найти хорошую последовательность индексов с **максимальной шириной**. Одноклассник обещал вам, что строки s и t таковы, что хотя бы одна хорошая последовательность точно существует.

Входные данные

Первая строка входных данных содержит два числа n и m ($2 \leq m \leq n \leq 2 \cdot 10^5$) — длины строк s и t соответственно.

Во второй строке входных данных задана строка s , состоящая из n строчных букв латинского алфавита.

В третьей строке задана строка t , состоящая из m строчных букв латинского алфавита.

Гарантируется, что существует хотя бы одна хорошая последовательность индексов.

Выходные данные

Выведите одно число — максимальную ширину хорошей последовательности.

Примеры

входные данные	выходные данные
5 3 abbbc abc	3
входные данные	Скопировать
входные данные	Скопировать
5 2 aaaaa aa	4
входные данные	Скопировать
5 5 abcdef abcdf	1
входные данные	Скопировать
2 2 ab ab	1
входные данные	Скопировать

Примечание

В первом примере существует две хороших последовательности с шириной 3: это $\{1, 2, 5\}$ и $\{1, 4, 5\}$.

Во втором примере хорошая последовательность максимальной ширины — это $\{1, 5\}$.

В третьем примере есть лишь одна хорошая последовательность — это $\{1, 2, 3, 4, 5\}$.



C. Идеальная клавиатура

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Поликарп хочет собрать свою собственную клавиатуру. Клавиатуры с несколькими рядами слишком сложны для него, его клавиатура будет состоять только из одного ряда, где все 26 строчных латинских букв будут расположены в определенном порядке.

Поликарп использует один и тот же пароль s на всех сайтах, где он зарегистрирован (это плохо, но ему все равно). Он хочет собрать клавиатуру, которая позволит набирать этот пароль очень легко. Он не любит двигать пальцами при вводе пароля, поэтому для каждой пары соседних символов в s они должны быть соседними на клавиатуре. Например, если пароль $abacaba$, то клавиатура $cabdefghi\dots$ идеальна, так как символы a и i с соседние на клавиатуре, и a и b соседние на клавиатуре.

Гарантируется, что в s нет двух соседних одинаковых символов, поэтому, например, `password` не может быть паролем (два символа S являются соседними).

Можете ли вы помочь Поликарпу с выбором идеальной клавиатуры, если это возможно?

Входные данные

Первая строка содержит одно целое число T ($1 \leq T \leq 1000$) — количество наборов тестовых данных.

Затем следуют T наборов тестовых данных, каждый из которых содержит одну строку s ($1 \leq |s| \leq 200$). s состоит только из строчных латинских букв. В s нет двух соседних одинаковых символов.

Выходные данные

Для каждого набора входных данных выведите результат в следующем виде:

- если невозможно собрать идеальную клавиатуру, выведите **NO** (в верхнем регистре, **это имеет значение в этой задаче**);
- в противном случае выведите **YES** (в верхнем регистре), а затем строку, состоящую из 26 строчных латинских букв — идеальную клавиатуру. Каждая латинская буква должна появиться в этой строке ровно один раз. Если ответов несколько, выведите любой из них.

Пример

входные данные	Скопировать
5 ababa codedoca abcd zxzytyz abcdefghijklmnopqrstuvwxyz	
выходные данные	Скопировать
YES bacdefghijklmnopqrstuvwxyz YES edocabfghijklmnopqrstuvwxyz NO YES xzytabcdefghijklmnopqrstuvwxyz NO	




[Kap6502](#) | [Выйти](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Шокирующая расстановка

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам дан массив a_1, a_2, \dots, a_n , состоящий из целых чисел, такой что $a_1 + a_2 + \dots + a_n = 0$.

Переставьте элементы массива a таким образом, чтобы выполнялось следующее условие:

$$\max_{1 \leq l \leq r \leq n} |a_l + a_{l+1} + \dots + a_r| < \max(a_1, a_2, \dots, a_n) - \min(a_1, a_2, \dots, a_n),$$

где $|x|$ обозначает модуль числа x .

Более формально, вам нужно выяснить, существует ли такая перестановка p_1, p_2, \dots, p_n , что для массива $a_{p_1}, a_{p_2}, \dots, a_{p_n}$ выполняется условие выше, и найти полученный массив.

Напомним, что массив p_1, p_2, \dots, p_n называется перестановкой, если для каждого числа x от 1 до n существует ровно одно i от 1 до n такое, что $p_i = x$.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 50\,000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 300\,000$) — длина массива a .

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — элементы массива a . Гарантируется, что сумма массива a равняется нулю, иными словами: $a_1 + a_2 + \dots + a_n = 0$.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит 300 000.

Выходные данные

Для каждого набора входных данных, если переставить элементы массива a требуемым образом невозможно, в единственной строке выведите «No».

Если же это возможно, в первой строке выведите «Yes», а далее в отдельной строке n чисел — элементы a_1, a_2, \dots, a_n , переставленные в нужном порядке $(a_{p_1}, a_{p_2}, \dots, a_{p_n})$.

Если возможных ответов несколько, вы можете вывести любой из них.

Пример

входные данные	Скопировать
<pre>7 4 3 4 -2 -5 5 2 2 2 -3 -3 8 -3 -3 1 1 1 1 1 1 3 0 1 -1 7 -3 4 3 4 -4 -4 0 1 0 7 -18 13 -18 -17 12 15 13</pre>	
выходные данные	Скопировать
	<pre>Yes -5 -2 3 4 Yes -3 2 -3 2 2 Yes 1 1 1 -3 1 1 1 -3 Yes -1 0 1 Yes 4 -4 4 -4 0 3 -3 No Yes</pre>

| 13 12 -18 15 -18 13 -17 |

Примечание

В первом наборе входных данных $\max(a_1, \dots, a_n) - \min(a_1, \dots, a_n) = 9$. Поэтому элементы можно переставить следующим образом: $[-5, -2, 3, 4]$. Несложно видеть, что для такой расстановки значение $|a_l + \dots + a_r|$ всегда не больше 7, а значит и меньше 9.

Во втором наборе входных данных можно переставить элементы массива следующим образом: $[-3, 2, -3, 2, 2]$. Тогда максимальный модуль суммы будет достигаться на подмассиве $[-3, 2, -3]$, и будет равен $|-3 + 2 + -3| = |-4| = 4$, что меньше чем 5.

В четвёртом тестовом примере любая перестановка массива a подойдёт в качестве ответа, в том числе и $[-1, 0, 1]$.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 03.11.2025 12:24:10_{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке




[Kap6502](#) | [Выйти](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Цветные порталы

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На прямой расположено n городов. Города пронумерованы целыми числами от 1 до n .

Для перемещения между городами используются порталы. Порталы бывают 4 цветов: синий, зеленый, красный и желтый. В каждом городе расположены порталы двух разных цветов. Из города i можно переместиться в город j , если в них есть порталы одинакового цвета (например, между городом «красный-синий» и городом «синий-зеленый» можно перемещаться). Данное перемещение стоит $|i - j|$ монет.

Ваша задача — ответить на q независимых запросов: посчитайте минимальную стоимость, чтобы переместиться из города x в город y .

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Первая строка каждого набора содержит два целых числа n и q ($1 \leq n, q \leq 2 \cdot 10^5$) — количество городов и количество запросов, соответственно.

Вторая строка содержит n строк одного из следующих типов: BG, BR, BY, GR, GY или RY; i -я из них описывает порталы, находящиеся в i -м городе; символ B обозначает, что в городе есть портал синего цвета, G — зеленого, R — красного, а Y — желтого.

j -я из следующих q строк содержит два целых числа x_j и y_j ($1 \leq x_j, y_j \leq n$) — описание j -го запроса.

Дополнительные ограничения на входные данные:

- сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$;
- сумма q по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого запроса выведите одно целое число — минимальную стоимость, чтобы переместиться из города x в город y (или -1 , если это невозможно).

Пример

входные данные	Скопировать
<pre>2 4 5 BR BR GY GR 1 2 3 1 4 4 1 4 4 2 2 1 BG RY 1 2</pre>	
выходные данные	Скопировать
<pre>1 4 0 3 2 -1</pre>	

[Kap6502](#) | [Выйти](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

E. Перестановка строк и столбцов

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

У вас была матрица a размера n на m , содержащая перестановку целых чисел от 1 до $n \cdot m$.

Перестановкой из n целых чисел называется массив, содержащий все числа от 1 до n ровно один раз. Например, массивы $[1], [2, 1, 3], [5, 4, 3, 2, 1]$ являются перестановками, а массивы $[1, 1], [100], [1, 2, 4, 5]$ — нет.

Матрица содержит перестановку, если при выписывании всех её элементов полученный массив является перестановкой. Матрицы $[[1, 2], [3, 4]], [[1]], [[1, 5, 3], [2, 6, 4]]$ содержат перестановки, а матрицы $[[2]], [[1, 1], [2, 2]], [[1, 2], [100, 200]]$ — нет.

За одну операцию вы можете совершить одно из двух следующих действий:

- выбрать столбец c и столбец d ($1 \leq c, d \leq m, c \neq d$) и поменять эти столбцы местами;
- выбрать строку c и строку d ($1 \leq c, d \leq n, c \neq d$) и поменять эти строки местами.

Вы можете совершить любое количество операций.

Вам даны исходная матрица a и матрица b . Ваша задача — определить, можно ли с помощью данных операций сделать из матрицы a матрицу b .

Входные данные

Первая строка содержит целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

Первая строка описания каждого набора входных данных содержит 2 целых числа n и m ($1 \leq n, m \leq n \cdot m \leq 2 \cdot 10^5$) — размеры матрицы.

Следующие n строк содержат по m целых чисел a_{ij} каждая ($1 \leq a_{ij} \leq n \cdot m$). Гарантируется, что матрица a является перестановкой.

Следующие n строк содержат по m целых чисел b_{ij} каждая ($1 \leq b_{ij} \leq n \cdot m$). Гарантируется, что матрица b является перестановкой.

Гарантируется, что сумма значений $n \cdot m$ по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите «YES», если вторая матрица может быть получена из первой, и «NO» в противном случае.

Вы можете выводить каждую букву в любом регистре (строчную или заглавную). Например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ.

Пример

Входные данные

[Скопировать](#)

```
7
1 1
1
1
2 2
1 2
3 4
4 3
2 1
2 2
1 2
3 4
4 3
1 2
3 4
1 5 9 6
12 10 4 8
7 11 3 2
1 5 9 6
12 10 4 8
7 11 3 2
3 3
```

```
1 5 9
6 4 2
3 8 7
9 5 1
2 4 6
7 8 3
2 3
1 2 6
5 4 3
6 1 2
3 4 5
1 5
5 1 2 3 4
4 2 5 1 3
```

Входные данные**Скопировать**

```
YES
YES
NO
YES
YES
NO
YES
```

Примечание

Во втором примере исходная матрица выглядит так:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

При обмене строк 1 и 2 местами она станет такой:

$$\begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}$$

При обмене столбцов 1 и 2 местами она станет равна матрице b :

$$\begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$$

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 03.11.2025 12:24:15^{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке




[Kap6502](#) | [Выйти](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Максимизация MEX

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Напомним, что **MEX** массива — это **минимальное неотрицательное целое число**, которое не представлено в массиве.

Примеры:

- для массива $[0, 0, 1, 0, 2]$ MEX равен 3, потому что числа 0, 1 и 2 представлены в массиве, а 3 является минимальным неотрицательным целым числом, не представленным в массиве;
- для массива $[1, 2, 3, 4]$ MEX равен 0, потому что 0 является минимальным неотрицательным целым числом, не представленным в массиве;
- для массива $[0, 1, 4, 3]$ MEX равен 2, потому что 2 является минимальным неотрицательным целым числом, не представленным в массиве.

Вам задан пустой массив $a = []$ (иными словами, массив нулевой длины). Также вам задано положительное целое число x .

Вам дано q запросов. j -й запрос состоит из одного целого числа y_j , которое означает, что вы должны дописать один элемент y_j в массив. Длина массива увеличивается на 1 после каждого запроса.

За один ход вы можете выбрать любой индекс i и присвоить $a_i := a_i + x$ или $a_i := a_i - x$ (т.е. увеличить или уменьшить любой элемент на x). Единственное ограничение — a_i не может стать отрицательным. Так как изначально массив пуст, вы можете совершать ходы только после первого запроса.

Вам нужно максимизировать **MEX** (minimum excluded) массива, применив какое-то количество подобных операций (вы также можете применить несколько таких операций к одному и тому же элементу).

Вам необходимо найти ответ после каждого из q запросов (то есть j -й ответ соответствует массиву длины j).

Операции отменяются после каждого запроса. То есть массив a после j -го запроса равен $[y_1, y_2, \dots, y_j]$.

Входные данные

Первая строка входных данных содержит два целых числа q, x ($1 \leq q, x \leq 4 \cdot 10^5$) — количество запросов и значение числа x .

Следующие q строк описывают запросы. j -й запрос состоит из целого числа y_j ($0 \leq y_j \leq 10^9$) и означает, что вы должны дописать один элемент y_j в массив.

Выходные данные

Выведите ответ на изначальную задачу после каждого запроса — для запроса j выведите максимальное значение **MEX** после первых j запросов. Стоит заметить, что запросы зависят (массив меняется после каждого запроса), но операции между запросами независимы.

Примеры

входные данные	Скопировать
7 3 0 1 2 2 0 0 10	
выходные данные	Скопировать
1 2 3 3 4 4 7	
входные данные	Скопировать
4 3 1 2	

1	
2	
Входные данные	Скопировать
0	
0	
0	
0	

Примечание

В первом тестовом примере:

- После первого запроса массив равен $a = [0]$: нет необходимости применять операции, максимально возможный MEX равен 1.
- После второго запроса массив равен $a = [0, 1]$: нет необходимости применять операции, максимально возможный MEX равен 2.
- После третьего запроса массив равен $a = [0, 1, 2]$: нет необходимости применять операции, максимально возможный MEX равен 3.
- После четвертого запроса массив равен $a = [0, 1, 2, 2]$: нет необходимости применять операции, максимально возможный MEX равен 3 (вы не можете сделать его больше, применяя операции).
- После пятого запроса массив равен $a = [0, 1, 2, 2, 0]$: вы можете совершить операцию $a[4] := a[4] + 3 = 3$. Массив изменится и станет равен $a = [0, 1, 2, 2, 3]$. Теперь MEX является максимально возможным и равен 4.
- После шестого запроса массив равен $a = [0, 1, 2, 2, 0, 0]$: вы можете совершить операцию $a[4] := a[4] + 3 = 0 + 3 = 3$. Массив изменится и станет равен $a = [0, 1, 2, 2, 3, 0]$. Теперь MEX является максимально возможным и равен 4.
- После седьмого запроса массив равен $a = [0, 1, 2, 2, 0, 0, 10]$. Вы можете совершить следующие операции:
 - $a[3] := a[3] + 3 = 2 + 3 = 5$,
 - $a[4] := a[4] + 3 = 0 + 3 = 3$,
 - $a[5] := a[5] + 3 = 0 + 3 = 3$,
 - $a[5] := a[5] + 3 = 3 + 3 = 6$,
 - $a[6] := a[6] - 3 = 10 - 3 = 7$,
 - $a[6] := a[6] - 3 = 7 - 3 = 4$.

Результирующий массив будет равен $a = [0, 1, 2, 5, 3, 6, 4]$. Теперь MEX является максимально возможным и равен 7.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:24:17 UTC+5 (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

[Kap6502](#) | [Выйти](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Сервал и формула

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам даны два целых положительных числа x и y ($1 \leq x, y \leq 10^9$).

Найдите целое **неотрицательное** число $k \leq 10^{18}$, такое что $(x + k) + (y + k) = (x + k) \oplus (y + k)$ верно*, или определите, что такое целое число не существует.

*⊕ обозначает операцию **побитового исключающего ИЛИ**.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Единственная строка каждого набора входных данных содержит два целых числа x и y ($1 \leq x, y \leq 10^9$) — заданные числа.

Выходные данные

Для каждого набора входных данных выведите одно целое число k ($0 \leq k \leq 10^{18}$) — найденное вами целое число. Выведите -1 , если искомого числа не существует.

Если существует несколько ответов, вы можете вывести любой из них.

Пример

входные данные	Скопировать
5 2 5 6 6 19 10 1024 4096 1198372 599188	
выходные данные	Скопировать
0 -1 1 1024 28	

Примечание

В первом наборе входных данных, поскольку $(2 + 0) + (5 + 0) = (2 + 0) \oplus (5 + 0) = 7$, $k = 0$ является верным ответом. Обратите внимание, что $k = 4$ также является верным ответом, потому что $(2 + 4) + (5 + 4) = (2 + 4) \oplus (5 + 4) = 15$.

Во втором наборе входных данных, $(x + k) \oplus (y + k) = (6 + k) \oplus (6 + k) = 0$. Однако, $(x + k) + (y + k) > 0$ выполняется для каждого $k \geq 0$, что подразумевает, что такое целое число k не существует.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:24:18^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ITMO

[Kap6502](#) | [Выходи](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Префиксы с нулевой суммой

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Красота массива v_1, v_2, \dots, v_n равняется количеству индексов i ($1 \leq i \leq n$), таких что $v_1 + v_2 + \dots + v_i = 0$.

У вас есть массив a_1, a_2, \dots, a_n длины n . Вы можете применить к нему следующую операцию сколько угодно раз:

- выбрать индекс i ($1 \leq i \leq n$), такой что $a_i = 0$;
- после этого заменить a_i на произвольное целое число.

Какой максимальной красоты массив a вы можете получить?

Входные данные

В первой строке задано одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют описания этих наборов.

В первой строке дано одно число n ($1 \leq n \leq 2 \cdot 10^5$) — длина массива a .

Во второй строке даны n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — массив a .

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно число: максимальную красоту массива a после применения к нему операций.

Пример

входные данные	Скопировать
5 5 2 0 1 -1 0 3 1000000000 1000000000 0 4 0 0 0 0 8 3 0 2 -10 10 -30 30 0 9 1 0 0 1 -1 0 1 0 -1	
выходные данные	Скопировать
3 1 4 4 5	

Примечание

В первом наборе входных данных вы можете заменить a_2 на -2 за одну операцию. После этого массив a станет равен $[2, -2, 1, -1, 0]$ и будет иметь красоту 3:

- $a_1 + a_2 = 2 - 2 = 0$;
- $a_1 + a_2 + a_3 + a_4 = 2 - 2 + 1 - 1 = 0$;
- $a_1 + a_2 + a_3 + a_4 + a_5 = 2 - 2 + 1 - 1 + 0 = 0$.

Во втором наборе входных данных вы можете заменить a_3 на $-2\ 000\ 000\ 000$, чтобы получить массив красоты 1.

В третьем наборе входных данных вы можете не делать операции.


[Kap6502](#) | [Выходи](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

E. Коля и кинотеатр

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Недавно Коля узнал, что скоро в его городе откроется новый кинотеатр, который будет показывать новый фильм каждый день на протяжении n дней. Так, в день номер $1 \leq i \leq n$, кинотеатр покажет премьеру i -го фильма. Коля узнал расписание фильмов и сопоставил каждому фильму его интересность — целую величину a_i .

Однако даже самые интересные фильмы не будут казаться такими, если фильмы долго не смотреть, поэтому интересность очередного фильма снизится на $d \cdot cnt$, где d — заранее определенная величина, а cnt — кол-во дней с последнего посещения кинотеатра. Также известно, что Коля успел посетить другой кинотеатр за день до открытия нового — в день номер 0. **Так, если мы впервые посетим кинотеатр в день с номером i , то cnt — кол-во дней с последнего посещения кинотеатра будет равно i .**

Так, например, если $d = 2$, а $a = [3, 2, 5, 4, 6]$, то посетив фильмы с номерами 1 и 3, значение cnt для дня номер 1 будет равно $1 - 0 = 1$, а значение cnt для дня номер 3 будет равно $3 - 1 = 2$, так суммарная интересность фильмов будет $a_1 - d \cdot 1 + a_3 - d \cdot 2 = 3 - 2 \cdot 1 + 5 - 2 \cdot 2 = 2$.

К несчастью, у Коли есть время только на то, чтобы посетить **не более m фильмов**. Помогите ему составить план посещения кинотеатра так, чтобы суммарная интересность всех посещенных им фильмов была максимальна.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится единственное целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит три целых числа n, m и d ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq n$, $1 \leq d \leq 10^9$).

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — интересность фильмов.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите единственное число — максимальную суммарную интересность просмотренных фильмов.

Пример

входные данные	Скопировать
6 5 2 2 3 2 5 4 6 4 3 2 1 1 1 1 6 6 6 -82 45 1 -77 39 11 5 2 2 3 2 5 4 8 2 1 1 -1 2 6 3 2 -8 8 -2 -1 9 0	
выходные данные	Скопировать
2 0 60 3 0 7	

Примечание

Первый пример разобран в условии.

Во втором примере, оптимально не посещать никакие фильмы.

В третьем наборе входных данных оптимально посетить фильмы с номерами 2, 3, 5, 6, тогда суммарная интересность


[Kap6502](#) | [Выход](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

Н. Третья буква

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Чтобы выиграть свою самую трудную битву, Мирча придумал отличную стратегию для своей армии. У него есть n солдат, и он решил расположить их определенным образом в лагерях. Каждый солдат должен принадлежать ровно одному лагерю, и на каждой целочисленной точке оси x находится один лагерь (на точках $\dots, -2, -1, 0, 1, 2, \dots$).

Стратегия состоит из m условий. Условие i гласит, что солдат a_i должен принадлежать лагерю, который находится в d_i метрах перед лагерем, к которому принадлежит человек b_i . (Если $d_i < 0$, то лагерь a_i должен быть в $-d_i$ метрах позади лагеря b_i .)

Теперь Мирча задается вопросом, существует ли разбиение солдат, которое удовлетворяет условию, и просит вашей помощи! Ответьте «YES», если существует разбиение из n солдат, которое удовлетворяет **всем** m условиям, и «NO» в противном случае.

Обратите внимание, что два разных солдата **могут** быть размещены в одном лагере.

Входные данные

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

Первая строка каждого набора содержит два положительных целых числа n и m ($2 \leq n \leq 2 \cdot 10^5$; $1 \leq m \leq n$) — количество солдат и количество условий соответственно.

Затем следуют m строк, каждая из которых содержит 3 целых числа: a_i, b_i, d_i ($a_i \neq b_i$; $1 \leq a_i, b_i \leq n$; $-10^9 \leq d_i \leq 10^9$) — описывающие условия, объясненные в условии. Обратите внимание, что если d_i положительно, то a_i должен быть в d_i метрах перед b_i , а если отрицательно, то a_i должен быть в $-d_i$ метрах позади b_i .

Обратите внимание, что сумма n по всем наборам входных данных не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите «YES», если существует такое расположение n солдат, которое удовлетворяет **всем** m условиям, и «NO» в противном случае.

Пример

входные данные	Скопировать
<pre>4 5 3 1 2 2 2 3 4 4 2 -6 6 5 1 2 2 2 3 4 4 2 -6 5 4 4 3 5 100 2 2 1 2 5 1 2 4 4 1 1 2 3</pre>	
выходные данные	Скопировать
<pre>YES NO NO YES</pre>	

Примечание

В первом примере мы можем разделить солдат на лагеря следующим образом:

- Солдат 1 в лагере с координатой $x = 3$.
- Солдат 2 в лагере с координатой $x = 5$.
- Солдат 3 в лагере с координатой $x = 9$.
- Солдат 4 в лагере с координатой $x = 11$.

Для второго примера не существует разбиения, которое может удовлетворить все ограничения одновременно.


[Kap6502](#) | [Выходи](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

E. Оглянись

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Вам дан массив целых чисел a_1, a_2, \dots, a_n . Вам нужно сделать его неубывающим за наименьшее количество операций. За одну операцию вы делаете следующее:

- Выберите индекс $1 \leq i \leq n$,
- Сделайте $a_i = a_i \cdot 2$.

Массив b_1, b_2, \dots, b_n является неубывающим, если $b_i \leq b_{i+1}$ для всех $1 \leq i < n$.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит единственное целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует их описание.

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

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите наименьшее количество операций, которое нужно сделать, чтобы массив стал неубывающим.

Пример

входные данные	Скопировать
<pre>9 1 1 2 2 1 3 3 2 1 4 7 1 5 3 5 11 2 15 7 10 6 1 8 2 16 8 16 2 624323799 708290323 12 2 1 1 3 3 11 12 22 45 777 777 1500 12 12 11 10 9 8 7 6 5 4 3 2 1</pre>	
выходные данные	Скопировать
<pre>0 1 3 6 10 3 0 2 66</pre>	

Примечание

В первом наборе входных данных операции не нужно делать.

Во втором наборе входных данных нужно выбрать $i = 2$, после чего массив будет равен $[2, 2]$.

В третьем наборе входных данных можно применить следующие операции:

- Выбрать $i = 3$, после чего массив будет равен $[3, 2, 2]$,
- Выбрать $i = 3$, после чего массив будет равен $[3, 2, 4]$,


[Kap6502](#) | [Выходи](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

E. Бинарный поиск

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Антону стало скучно в походе, и он захотел что-нибудь порешать. Он спросил у Кирилла, нет ли у него какой-то новой задачи, и у Кирилла она естественно нашлась.

Дана перестановка p размера n , а также число x , которое необходимо найти. Перестановкой длины n является массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — перестановка, но $[1, 2, 2]$ не перестановка (2 встречается в массиве дважды) и $[1, 3, 4]$ тоже не перестановка ($n = 3$, но в массиве встречается 4).

Вы решили, что вы крутой программист, поэтому будете использовать для поиска продвинутый алгоритм — бинарный поиск. Однако, вы забыли, что для бинарного поиска массив должен быть отсортирован.

Вы не отчаялись и решили все равно применить этот алгоритм, а чтобы получить правильный ответ, вы перед запуском алгоритма можете **не более 2** раз выполнить следующую операцию: выбрать индексы i, j ($1 \leq i, j \leq n$) и поменять местами элементы на позициях i и j .

После этого выполняется бинарный поиск. В начале алгоритма объявляются две переменные $l = 1$ и $r = n + 1$. Далее выполняется цикл:

1. Если $r - l = 1$, закончить цикл
2. $m = \lfloor \frac{r+l}{2} \rfloor$
3. Если $p_m \leq x$, присвоить $l = m$, иначе $r = m$.

Цель — до алгоритма переставить числа в перестановке так, чтобы после выполнения алгоритма p_l было равно x . Можно показать, что 2 операции всегда достаточно.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 2 \cdot 10^4$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

Первая строка каждого набора содержит два целых числа n и x ($1 \leq x \leq n \leq 2 \cdot 10^5$) — длину перестановки и число, которое нужно найти.

Во второй строке через пробел вводится перестановка p ($1 \leq p_i \leq n$).

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите в первой строке целое число k ($0 \leq k \leq 2$) — количество совершенных вами операций. В следующих k строках через пробел выведите по 2 целых числа i, j ($1 \leq i, j \leq n$) означающих, что вы меняете местами элементы на позициях i и j .

Обратите внимание, что вам не нужно минимизировать количество операций.

Пример

входные данные	Скопировать
5 6 3 1 2 3 4 5 6 6 5 3 1 6 5 2 4 5 1 3 5 4 2 1 6 3 4 3 1 5 2 6 3 2 3 2 1	
выходные данные	Скопировать
0 1 3 4 2 2 4	

1	5
2	
4	5
2	4
1	
1	3

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:24:46_{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке




[Kap6502](#) | [Выйти](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

B. Фейковые пластиковые деревья

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Нам дано корневое дерево, состоящее из n вершин, пронумерованных от 1 до n . Корнем дерева является вершина 1, а родителем вершины v является p_v .

В каждой вершине записано число, изначально все числа равны 0. Обозначим число, записанное в вершине v , как a_v .

Для каждой v мы хотим, чтобы a_v находилось между l_v и r_v ($l_v \leq a_v \leq r_v$).

За одну операцию мы делаем следующее:

- Выбираем некоторую вершину v . Пусть b_1, b_2, \dots, b_k — вершины на пути от вершины 1 до вершины v (то есть $b_1 = 1$, $b_k = v$ и $b_i = p_{b_{i+1}}$).
- Выберем неубывающий массив c длины k из неотрицательных целых чисел: $0 \leq c_1 \leq c_2 \leq \dots \leq c_k$.
- Для каждого i ($1 \leq i \leq k$), увеличим a_{b_i} на c_i .

Какое минимальное количество операций необходимо для достижения нашей цели?

Входные данные

Первая строка содержит целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество вершин в дереве.

Вторая строка каждого набора входных данных содержит $n - 1$ целых чисел, p_2, p_3, \dots, p_n ($1 \leq p_i < i$), где p_i обозначает родителя вершины i .

В i -й из следующих n строк содержится два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq 10^9$).

Гарантируется, что сумма n по всем наборам входных данных не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите минимально необходимое количество операций.

Пример

входные данные	Скопировать
4	
2	
1	
1 5	
2 9	
3	
1 1	
4 5	
2 4	
6 10	
4	
1 2 1	
6 9	
5 6	
4 5	
2 4	
5	
1 2 3 4	
5 5	
4 4	
3 3	
2 2	
1 1	
выходные данные	Скопировать
1	
2	
2	
5	

Примечание

В первом наборе входных данных мы можем достичь цели с помощью одной операции: выбрать $v = 2$ и $c = [1, 2]$, в результате чего $a_1 = 1, a_2 = 2$.

Во втором наборе входных данных мы можем достичь цели с помощью двух операций: сначала выбираем $v = 2$ и $c = [3, 3]$, в результате чего $a_1 = 3, a_2 = 3, a_3 = 0$. Затем выбираем $v = 3, c = [2, 7]$, в результате $a_1 = 5, a_2 = 3, a_3 = 7$.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:24:47^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

[Kap6502](#) | [Выход](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

F. Уравнение мульти множества

ограничение по времени на тест: 4 секунды

ограничение по памяти на тест: 256 мегабайт

Мульти множества — это такой набор чисел, в котором могут быть равные элементы, а порядок чисел не имеет значение. Два мульти множества равны, когда каждое значение встречается в них одинаковое количество раз. Например, мульти множества $\{2, 2, 4\}$ и $\{2, 4, 2\}$ равны, но мульти множества $\{1, 2, 2\}$ и $\{1, 1, 2\}$ — нет.

Вам даны два мульти множества a и b , каждое состоит из n целых чисел.

За одну операцию можно любой элемент мульти множества b увеличить в два раза или уменьшить в два раза (с округлением вниз). Другими словами, вам доступна одна из следующих операций для элемента x из мульти множества b :

- либо заменить x на $x \cdot 2$,
- либо заменить x на $\lfloor \frac{x}{2} \rfloor$ (округление вниз).

Обратите внимание, что изменять элементы мульти множества a нельзя.

Проверьте, можно ли за произвольное количество операций (возможно, 0) сделать так, чтобы мульти множество b стало равно мульти множеству a .

Например, если $n = 4$, $a = \{4, 24, 5, 2\}$, $b = \{4, 1, 6, 11\}$, то ответ положительный. Можно действовать следующим образом:

- Заменим 1 на $1 \cdot 2 = 2$. Получим $b = \{4, 2, 6, 11\}$.
- Заменим 11 на $\lfloor \frac{11}{2} \rfloor = 5$. Получим $b = \{4, 2, 6, 5\}$.
- Заменим 6 на $6 \cdot 2 = 12$. Получим $b = \{4, 2, 12, 5\}$.
- Заменим 12 на $12 \cdot 2 = 24$. Получим $b = \{4, 2, 24, 5\}$.
- получили равные мульти множества $a = \{4, 24, 5, 2\}$ и $b = \{4, 2, 24, 5\}$.

Входные данные

В первой строке входных данных задано единственное целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Каждый набор входных данных состоит из трех строк.

В первой строке набора задано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество элементов в мульти множествах a и b .

Во второй строке даны n целых чисел: a_1, a_2, \dots, a_n ($1 \leq a_1 \leq a_2 \leq \dots \leq a_n \leq 10^9$) — элементы набора a . Обратите внимание, что среди элементов могут быть равные.

В третьей строке даны n целых чисел: b_1, b_2, \dots, b_n ($1 \leq b_1 \leq b_2 \leq \dots \leq b_n \leq 10^9$) — элементы набора b . Обратите внимание, что среди элементов могут быть равные.

Гарантируется, что сумма значений n по всем наборам входных данных не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных в отдельной строке выведите:

- YES, если можно сделать так, чтобы мульти множество b стало равно a ,
- NO в противном случае.

Вы можете выводить YES и NO в любом регистре (например, строки yEs, yes, Yes и YES будут распознаны как положительный ответ).

Пример

входные данные	Скопировать
5 4 2 4 5 24 1 4 6 11 3 1 4 17 4 5 31 5 4 7 10 13 14 2 14 14 26 42 5	

```
2 2 4 4 4
28 46 62 71 98
6
1 2 10 16 64 80
20 43 60 74 85 99
```

выходные данные**Скопировать**

```
YES
NO
YES
YES
YES
```

Примечание

Первый пример разобран в условии.

Во втором примере невозможно получить значение 31 из чисел мульти множества b доступными операциями.

В третьем примере можно действовать следующим образом:

- Заменим 2 на $2 \cdot 2 = 4$. Получим $b = \{4, 14, 14, 26, 42\}$.
- Заменим 14 на $\lfloor \frac{14}{2} \rfloor = 7$. Получим $b = \{4, 7, 14, 26, 42\}$.
- Заменим 26 на $\lfloor \frac{26}{2} \rfloor = 13$. Получим $b = \{4, 7, 14, 13, 42\}$.
- Заменим 42 на $\lfloor \frac{42}{2} \rfloor = 21$. Получим $b = \{4, 7, 14, 13, 21\}$.
- Заменим 21 на $\lfloor \frac{21}{2} \rfloor = 10$. Получим $b = \{4, 7, 14, 13, 10\}$.
- получили равные мульти множества $a = \{4, 7, 10, 13, 14\}$ и $b = \{4, 7, 14, 13, 10\}$.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:24:49 UTC+5 (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

D. Удаление строки

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам дана строка s , состоящая из n символов. Каждый символ — либо 0, либо 1.

Вы можете проводить операции со строкой. Каждая операция состоит из двух шагов:

1. выбрать целое число i от 1 до длины строки s , после чего удалить символ s_i (длина строки уменьшается на 1, номера символов правее удаленного тоже уменьшаются на 1);
2. если строка s не является пустой, удалить максимальный по длине префикс, состоящий из одинаковых символов (номера остальных символов и длина строки уменьшаются на длину удаленного префикса).

Обратите внимание, что в каждой операции оба шага обязательны, и их порядок нельзя менять.

Например, если у вас есть строка $s = 111010$, первая операция может быть одной из следующих:

1. выбрать $i = 1$: тогда мы получим $111010 \rightarrow 11010 \rightarrow 010$;
2. выбрать $i = 2$: тогда мы получим $111010 \rightarrow 11010 \rightarrow 010$;
3. выбрать $i = 3$: тогда мы получим $111010 \rightarrow 11010 \rightarrow 010$;
4. выбрать $i = 4$: тогда мы получим $111010 \rightarrow 11110 \rightarrow 0$;
5. выбрать $i = 5$: тогда мы получим $111010 \rightarrow 11100 \rightarrow 00$;
6. выбрать $i = 6$: тогда мы получим $111010 \rightarrow 11101 \rightarrow 01$.

Вы заканчиваете проводить операции, когда строка s становится пустой. Какое максимальное количество операций вы можете провести?

Входные данные

В первой строке задано одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

В первой строке каждого набора задано одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина строки s .

Во второй строке задана s — строка из n символов. Каждый символ — либо 0, либо 1.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных, выведите одно целое число — максимальное количество операций, которые вы можете провести.

Пример

входные данные	Скопировать
5 6 111010 1 0 1 1 2 11 6 101010	
выходные данные	Скопировать
3 1 1 1 3	

Примечание

В первом наборе входных данных, мы можем, например, выбрать $i = 2$ и получить строку 010 после первой операции. После этого, выбрать $i = 3$ и получить строку 1. Наконец, мы можем выбрать только $i = 1$ и получить пустую строку.

[Kap6502](#) | [Выходи](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Пенчик и пустынный кролик

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Пенчик бросил себе вызов — выжить под полуденным солнцем в Аравийской пустыне!

На линейном оазисе в ряд растёт n пальм, обозначим их высоты a_i . Во время прогулки Пенчик заметил пустынного кролика, который готовится прыгнуть вдоль линии пальм.

Кролик может перепрыгнуть с i -го дерева на j -е, если верно ровно одно из следующих условий:

- $j < i$ и $a_j > a_i$: кролик может прыгнуть назад на более высокое дерево.
- $j > i$ и $a_j < a_i$: кролик может прыгнуть вперёд на более низкое дерево.

Для каждого i от 1 до n определите максимальную высоту среди всех деревьев, которые может достичь кролик, если он начнёт с i -го дерева.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t ($1 \leq t \leq 5 \cdot 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 5 \cdot 10^5$) — количество деревьев.

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — высоты деревьев.

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

Выходные данные

Для каждого набора входных данных выведите n целых чисел. i -е число должно равняться максимальной высоте среди всех деревьев, которые может достичь кролик, если он начнёт с дерева i .

Пример

входные данные	Скопировать
5	
4	
2 3 1 4	
5	
5 4 3 2 1	
4	
2 1 1 3	
4	
1 1 3 1	
8	
2 4 1 6 3 8 5 7	
выходные данные	Скопировать
3 3 3 4	
5 5 5 5	
2 2 2 3	
1 1 3 3	
8 8 8 8 8 8 8 8	

Примечание

В первом наборе входных данных высоты деревьев равны $a = [2, 3, 1, 4]$.

- Если кролик стартует с первого дерева, то он может перепрыгнуть на третье дерево, так как $3 > 1$ и $1 < 2$. Затем кролик может перепрыгнуть на второе дерево, так как $2 < 3$ и $3 > 1$. Можно доказать, что кролик не может допрыгнуть до четвёртого дерева; следовательно, максимальная высота дерева, до которого может допрыгнуть кролик, равна $a_2 = 3$.
- Если кролик стартует с четвёртого дерева, ему не нужно никуда прыгать, так как он уже находится на самом высоком дереве.

Во втором наборе входных данных кролик может прыгнуть на первое дерево независимо от того, с какого дерева он стартует.

В пятом наборе входных данных, если кролик начинает с пятого дерева, он может прыгнуть на четвёртое дерево. Затем кролик может перепрыгнуть на седьмое дерево и, наконец, достичь шестого дерева. Таким образом, максимальная высота дерева, до которого может добраться кролик, равна 8.


[Kap6502](#) | [Выйти](#)
[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Магазин сладостей

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

В магазине продаётся n типов конфет с номерами от 1 до n . Одна конфета типа i стоит b_i монет. Всего в магазине есть a_i конфет типа i .

Вам необходимо расфасовать все конфеты по пачкам, каждая пачка должна состоять только из конфет одного типа. Формально, для каждого типа конфет i вам необходимо выбрать число d_i , обозначающее количество конфет типа i в одной пачке, так, чтобы a_i делилось без остатка на d_i .

Тогда стоимость одной пачки конфет типа i будет равна $b_i \cdot d_i$. Обозначим эту стоимость за c_i , то есть $c_i = b_i \cdot d_i$.

После расфасовки пачки конфет будут помещены на полку. Рассмотрим стоимости пачек, расположенных на полке, в порядке c_1, c_2, \dots, c_n . Для описания стоимостей будут использоваться ценники. Один ценник может описать стоимость всех товаров от l до r включительно, если $c_l = c_{l+1} = \dots = c_r$. Каждый из товаров от 1 до n должен быть описан хотя бы одним ценником. К примеру, если $c_1, \dots, c_n = [4, 4, 2, 4, 4]$, для описания всех товаров будет достаточно 3 ценника, первый ценник описывает товары 1, 2, второй: 3, третий: 4, 5.

Вам даны числа $a_1, b_1, a_2, b_2, \dots, a_n, b_n$. Ваша задача выбрать числа d_i так, чтобы a_i делилось на d_i для всех i , и необходимое количество ценников для описания стоимостей c_1, c_2, \dots, c_n было минимально возможным.

Для лучшего понимания условия ознакомьтесь с иллюстрацией первого набора входных данных первого теста:

 $a[1] = 20$ $b[1] = 3$ $d[1] = 4$ $c[1] = 12$	 $a[2] = 6$ $b[2] = 2$ $d[2] = 6$ $c[2] = 12$	 $a[3] = 14$ $b[3] = 5$ $d[3] = 7$ $c[3] = 35$	 $a[4] = 20$ $b[4] = 7$ $d[4] = 5$ $c[4] = 35$
12		35	

Повторим смысл используемых в задаче обозначений:

a_i — количество конфет типа i имеющихся в магазине.

b_i — стоимость одной конфеты типа i .

d_i — количество конфет типа i в одной пачке.

c_i — стоимость одной пачки конфет типа i , выражается по формуле $c_i = b_i \cdot d_i$.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 100\,000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($2 \leq n \leq 200\,000$) — количество видов конфет.

Каждая из следующих n строк каждого набора входных данных содержит два целых числа a_i и b_i ($1 \leq a_i \leq 10^9$, $1 \leq b_i \leq 10\,000$) — количество конфет и стоимость одной конфеты типа i соответственно.

Гарантируется, что сумма n по всем наборам входных данных не превышает 200 000.

Выходные данные

Для каждого набора входных данных, выведите минимальное количество ценников необходимое для описания стоимостей всех товаров в магазине.

Пример

входные данные

Скопировать

```
4
20 3
6 2
14 5
20 7
3
444 5
2002 10
2020 2
5
7 7
6 5
15 2
10 3
7 7
5
10 1
11 5
5 1
2 2
8 2
6
7 12
12 3
5 3
9 12
9 3
1000000000 10000
```

Входные данные

```
2
1
3
2
5
```

Примечание

В первом наборе входных данных можно выбрать $d_1 = 4$, $d_2 = 6$, $d_3 = 7$, $d_4 = 5$. Тогда стоимости товаров будут равны: [12, 12, 35, 35]. Для их описания хватят 2 ценников, первый ценник для c_1, c_2 и второй ценник для c_3, c_4 . Можно показать, что при любом корректном выборе d_i для описания всех товаров понадобится как минимум 2 ценника. Также обратите внимание, что этот пример иллюстрируется картинкой в условии задачи.

Во втором наборе входных данных при $d_1 = 4$, $d_2 = 2$, $d_3 = 10$ стоимости всех товаров будут равны 20. Таким образом 1 ценника хватит для описания всех товаров. Обратите внимание, что a_i делится на d_i для всех i , что является необходимым условием.

В третьем наборе входных данных нетрудно понять, что для описания 2, 3 и 4 товара может быть использован один ценник. И дополнительно понадобится по ценнику для товаров 1 и 5. Итого 3 ценника.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 03.11.2025 12:24:55^{UTC+5} (h1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ITMO