


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

A. Ключи от офиса

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На прямой есть n человек и k ключей. Каждый из людей хочет попасть в офис, который также расположен на прямой. Для этого каждому человеку нужно сначала дойти до точки, в которой находится ключ, взять его, а затем отправиться в офис. После того как человек взял ключ, больше этот ключ взять никто не сможет.

Перед вами стоит задача определить минимальное время по истечении которого все n людей смогут добраться до офиса с ключами. Считайте, что каждый из людей преодолевает единицу расстояния за 1 секунду. Если в точку с ключом одновременно придут два человека, то ключ может взять только один из них. Человек не обязан брать ключ, мимо которого проходит.

Входные данные

В первой строке следуют три целых числа n , k и p ($1 \leq n \leq 1\,000$, $n \leq k \leq 2\,000$, $1 \leq p \leq 10^9$) — количество людей, количество ключей и позиция офиса.

Во второй строке следуют n различных целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — позиции, в которых изначально находятся люди. Позиции заданы в произвольном порядке.

В третьей строке следуют k различных целых чисел b_1, b_2, \dots, b_k ($1 \leq b_j \leq 10^9$) — позиции, в которых изначально находятся ключи. Позиции заданы в произвольном порядке.

Обратите внимание, что изначально в каждой точке может находиться не более одного человека и не более одного ключа. Человек и ключ могут находиться в одной точке.

Выходные данные

Выведите минимальное время в секундах по истечении которого все n людей смогут добраться до офиса с ключами.

Примеры

входные данные	Скопировать
2 4 50 20 100 60 10 40 80	
выходные данные	Скопировать
50	
входные данные	Скопировать
1 2 10 11 15 7	
выходные данные	Скопировать
7	

Примечание

В первом примере человек, находящийся в точке 20 должен взять ключ, находящийся в точке 40 и дойти с ним до офиса, который находится в точке 50. На это он потратит 30 секунд. Человек, находящийся в точке 100 может взять ключ, находящийся в точке 80 и дойти с ним до офиса. На это он потратит 50 секунд. Таким образом, через 50 секунд все люди могут оказаться в офисе с ключами.




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

E. Камень, ножницы, бумага

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Аня и Боря решили сыграть в игру «Камень, ножницы, бумага».

Игра состоит из раундов, каждый из раундов проводится независимо от остальных. В каждом раунде оба игрока одновременно показывают один из трех предметов: камень, ножницы или бумагу. При этом, если оба игрока показали одинаковые предметы, то в раунде объявляется ничья. В противном случае, действуют следующие правила:

- если один игрок показал камень, а другой ножницы, то игрок, показавший камень, объявляется победителем раунда, а другой игрок — проигравшим;
- если один игрок показал ножницы, а другой бумагу, то игрок, показавший ножницы, объявляется победителем раунда, а другой игрок — проигравшим;
- если один игрок показал бумагу, а другой камень, то игрок, показавший бумагу, объявляется победителем раунда, а другой игрок — проигравшим.

Аня и Боря решили, что они сыграют ровно n раундов описанной игры. Аня решила, что она a_1 раз покажет камень, a_2 раз покажет ножницы и a_3 раз покажет бумагу. Боря решил, что он b_1 раз покажет камень, b_2 раз покажет ножницы и b_3 раз покажет бумагу. При этом ни Аня, ни Боря еще **не выбрали** последовательность, в которой будут показывать предметы.

Гарантируется, что $a_1 + a_2 + a_3 = n$ и $b_1 + b_2 + b_3 = n$.

Перед вами стоит задача определить два числа:

1. минимальное количество раундов, которое может выиграть Аня;
2. максимальное количество раундов, которое может выиграть Аня.

Входные данные

В первой строке следует целое число n ($1 \leq n \leq 10^9$) — количество раундов.

Во второй строке следуют три целых числа a_1, a_2, a_3 ($0 \leq a_i \leq n$) — количество раз, которое Аня покажет камень, ножницы и бумагу, соответственно. Гарантируется, что $a_1 + a_2 + a_3 = n$.

В третьей строке следуют три целых числа b_1, b_2, b_3 ($0 \leq b_j \leq n$) — количество раз, которое Боря покажет камень, ножницы и бумагу, соответственно. Гарантируется, что $b_1 + b_2 + b_3 = n$.

Выходные данные

Выведите два целых числа — минимальное и максимальное количество раундов, которое может выиграть Аня.

Примеры

входные данные	Скопировать
2 0 1 1 1 1 0	
выходные данные	Скопировать
0 1	
входные данные	Скопировать
15 5 5 5 5 5 5	
выходные данные	Скопировать
0 15	
входные данные	Скопировать
3 0 0 3 3 0 0	
выходные данные	Скопировать
3 3	

Входные данные	Скопировать
686 479 178 29 11 145 530	
Выходные данные	Скопировать
22 334	

Входные данные	Скопировать
319 10 53 256 182 103 34	
Выходные данные	Скопировать
119 226	

Примечание

В первом примере Аня может не выиграть ни одного раунда, если покажет, например, сначала ножницы, а потом бумагу, а Боря покажет камень, а потом ножницы. В лучшем случае Аня выиграет один раунд, если, например, сначала покажет бумагу, а потом ножницы, а Боря покажет камень, а потом ножницы.

Во втором примере Аня может не выиграть ни одного раунда, если, например, Боря в каждом раунде будет показывать те же предметы, что и Аня.

В третьем примере Аня всегда показывает бумагу, а Боря всегда показывает камень, поэтому Аня в любом случае выиграет все три раунда.

[Codeforces](#) (с) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 18.11.2025 11:16:19_{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

D. Хранительница Бесконечного Зоопарка

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

В Сингапурском зоопарке появилась новая достопримечательность: Бесконечный Зоопарк.

Бесконечный Зоопарк может быть представлен графом с бесконечным числом вершин, обозначенных $1, 2, 3, \dots$. От вершины u до вершины $u + v$ существует ориентированное ребро, если и только если $u \& v = v$, где $\&$ обозначает операцию [побитового И](#). Других ребер в графе нет.

Хранительница Зоопарка получила q запросов. В i -м запросе она спросит вас, может ли она добраться от вершины u_i до вершины v_i по ориентированным ребрам.

Входные данные

Первая строка содержит целое число q ($1 \leq q \leq 10^5$) — количество запросов.

i -я из следующих q строк будет содержать два целых числа u_i, v_i ($1 \leq u_i, v_i < 2^{30}$) — запрос, сделанный Хранительницей Зоопарка.

Выходные данные

Для i -го из q запросов выведите «YES» в единственной строке, если Хранительница Зоопарка может добраться от вершины u_i до вершины v_i . В противном случае выведите «NO».

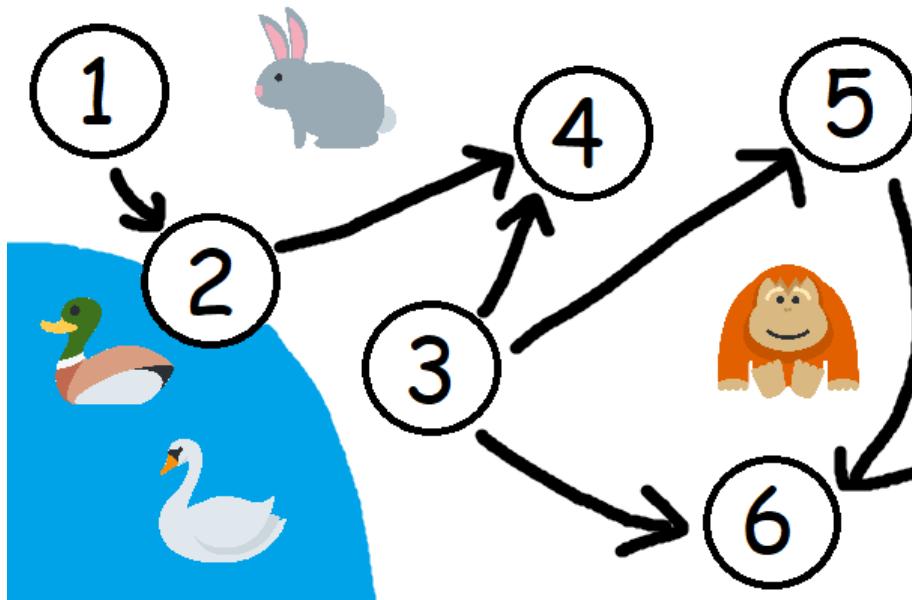
Ответ можно вывести в любом регистре. Например, если ответ «YES», то вывод «Yes» или «yeS» также будет считаться правильным ответом.

Пример

входные данные	Скопировать
5 1 4 3 6 1 6 6 2 5 5	
выходные данные	Скопировать
YES YES NO NO YES	

Примечание

Ниже показан подграф на вершинах $1, 2, 3, 4, 5, 6$.



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

F. Клад из отрезков

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Поликарп нашёл на улице n отрезков. Отрезок с номером i характеризуется двумя целыми числами l_i и r_i — координаты начала и конца отрезка, соответственно. Поликарп понял, что ему нужны не все отрезки, поэтому он хочет удалить некоторые из них.

Поликарп считает, что набор из k отрезков хороший, если существует отрезок $[l_i, r_i]$ ($1 \leq i \leq k$) из набора, такой что он пересекает каждый отрезок из набора (пересечение должно быть **точкой или отрезком**). Например, набор из 3 отрезков $[[1, 4], [2, 3], [3, 6]]$ является хорошим, так как отрезок $[2, 3]$ пересекает каждый отрезок из набора. Набор из 4 отрезков $[[1, 2], [2, 3], [3, 5], [4, 5]]$ хорошим не является.

Поликарпу интересно, какое минимальное количество отрезков ему надо удалить, чтобы набор из оставшихся отрезков стал хорошим?

Входные данные

В первой строке находится одно целое число t ($1 \leq t \leq 2 \cdot 10^5$) — количество наборов входных данных. Далее следуют t наборов входных данных.

В первой строке каждого набора входных данных находится одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество отрезков в наборе. Далее следуют n строк с описанием отрезков.

Каждый отрезок описывается двумя целыми числами l и r ($1 \leq l \leq r \leq 10^9$) — координаты начала и конца отрезка, соответственно.

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 — минимальное количество отрезков, которые необходимо удалить, чтобы набор оставшихся отрезков стал хорошим.

Пример

входные данные	Скопировать
4 3 1 4 2 3 3 6 4 1 2 2 3 3 5 4 5 5 1 2 3 8 4 5 6 7 9 10 5 1 5 2 4 3 5 3 8 4 8	
выходные данные	Скопировать
0 1 2 0	

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

D. Покупка префиксов

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

У вас есть массив a размера n , изначально заполненный нулями ($a_1 = a_2 = \dots = a_n = 0$). Также у вас есть массив целых чисел c размера n .

Изначально у вас есть k монет. Заплатив c_i монет, вы можете прибавить 1 ко всем элементам массива a с первого по i -й ($a_j + 1$ для всех $1 \leq j \leq i$). Покупать любое c_i можно произвольное число раз. Покупка возможна только при $k \geq c_i$, то есть в любой момент должно выполняться $k \geq 0$.

Найдите лексикографически наибольший массив a , который возможно получить.

Массив a лексикографически меньше массива b такой же длины, если и только если в первой позиции, где a и b различны, в массиве a элемент меньше, чем соответствующий элемент в b .

Входные данные

Первая строка содержит одно 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 одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — размер массивов a и c .

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

Третья строка каждого набора входных данных содержит одно целое число k ($1 \leq k \leq 10^9$) — количество монет, которое у вас есть.

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

Выходные данные

На каждый набор входных данных выведите n целых чисел a_1, a_2, \dots, a_n — лексикографически наибольший массив a , который можно получить.

Пример

входные данные	Скопировать
4 3 1 2 3 5 2 3 4 7 3 3 2 1 2 6 10 6 4 6 3 4 7	
выходные данные	Скопировать
5 0 0 2 1 2 2 2 2 2 2 2 2 1	

Примечание

В первом наборе входных данных a_1 не может быть больше 5, а если пять раз купить c_1 , то у нас не останется денег, поэтому $a = [5, 0, 0]$.

Во втором наборе входных данных a_1 не может быть больше 2, при этом мы можем купить c_1 и c_2 по одному разу (купить c_2 дважды не получится), поэтому $a = [2, 1]$.

[Kap6502](#) | [Выходи](#)

ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

B. Отсутствующая сумма подпоследовательности

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Даны два целых числа n и k . Найдите последовательность a неотрицательных целых чисел размером не более 25, такую, что выполняются следующие условия.

- Нет подпоследовательности a с суммой k .
- Для всех $1 \leq v \leq n$, где $v \neq k$, существует подпоследовательность a с суммой v .

Последовательность b является подпоследовательностью a , если b может быть получена из a путем удаления нескольких (возможно, нуля или всех) элементов, не изменяя порядок оставшихся элементов. Например, $[5, 2, 3]$ является подпоследовательностью $[1, 5, 7, 8, 2, 4, 3]$.

Можно показать, что при заданных ограничениях всегда существует решение.

Входные данные

Первая строка ввода содержит одно целое число t ($1 \leq t \leq 1000$) — количество тестов. Затем следует описание самих тестов.

Каждый тест состоит из одной строки, содержащей два целых числа n и k ($2 \leq n \leq 10^6$, $1 \leq k \leq n$) — описанные выше параметры.

Гарантируется, что сумма n по всем тестам не превышает 10^7 .

Выходные данные

Первая строка вывода для каждого теста должна содержать одно целое число m ($1 \leq m \leq 25$) — размер выбранной вами последовательности.

Вторая строка вывода для каждого теста должна содержать m целых чисел a_i ($0 \leq a_i \leq 10^9$) — элементы вашей выбранной последовательности.

Если существует несколько решений, выведите любое из них.

Пример

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

Примечание

В первом примере нам просто нужна подпоследовательность, дающая в сумме 1, но не дающая в сумме 2. Таким образом, массив $a = [1]$ подходит.

Во втором примере все элементы больше $k = 1$, поэтому ни одна подпоследовательность не дает в сумме 1. Каждое другое целое число между 1 и n присутствует в массиве, поэтому существует подпоследовательность размера 1, дающая в сумме каждое из этих чисел.

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

D. Андрей и побег из Капиграда

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

В Капиграде, столице Тяголяндии, произошло происшествие, все калибры в городе одичали и стали кидаться мандаринами. Андрей был вынужден сбежать из города максимально далеко, пользуясь порталами.

Тяголяндия представляет из себя числовую прямую, а Капиград находится в точке 0. По всей Тяголяндии расставлены n порталов, каждый из которых характеризуется четырьмя целыми числами l_i, r_i, a_i и b_i ($1 \leq l_i \leq a_i \leq b_i \leq r_i \leq 10^9$).

Обратите внимание, что отрезок $[a_i, b_i]$ **содержится** в отрезке $[l_i, r_i]$.

Если Андрей находится на отрезке $[l_i, r_i]$, тогда портал может его телепортировать в любую точку на отрезке $[a_i, b_i]$. У Андрея есть проездной, который позволяет ему пользоваться порталами неограниченное количество раз.

Андрей считает, что точка x находится на отрезке $[l, r]$, если выполняется неравенство $l \leq x \leq r$.

У Андрея есть q вариантов откуда начать свой побег, каждый вариант характеризуется одним целым числом x_i — позицией начала побега. Он хочет убежать от Капиграда как можно дальше (в точку с **максимально** возможной координатой). Помогите Андрею узнать, насколько далеко от Капиграда он может убежать, начиная в каждой из q позиций.

Входные данные

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

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

Каждая из следующих n строк содержит четыре целых числа l_i, r_i, a_i и b_i ($1 \leq l_i \leq a_i \leq b_i \leq r_i \leq 10^9$) — характеристики порталов.

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

В следующей строке задано q целых чисел x_1, x_2, \dots, x_q ($1 \leq x_i \leq 10^9$) — позиция, откуда Андрей начнет свой побег в i -м варианте.

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

Выходные данные

Для каждого набора входных данных выведите одну строку из q целых чисел, содержащую ответы на интересующие Андрея вопросы.

Пример

входные данные	Скопировать
5 3 6 17 7 14 1 12 3 8 16 24 20 22 6 10 2 23 15 28 18 3 3 14 7 10 16 24 20 22 1 16 3 14 9 2 4 6 8 18 23 11 13 15 2 1 4 2 3 3 9 6 7 3 4 8 1 5 18 24 18 24 1 8 2 4 11 16 14 14 26 32 28 30 5 10 6 8 9 15 14 13 27 22 17 31 1 7 6	

```

9 22 14 20
11 26 13 24
21 33 22 23
21 33 25 32
1 6 3 4
18 29 20 21
8
11 23 16 5 8 33 2 21

```

входные данные**Скопировать**

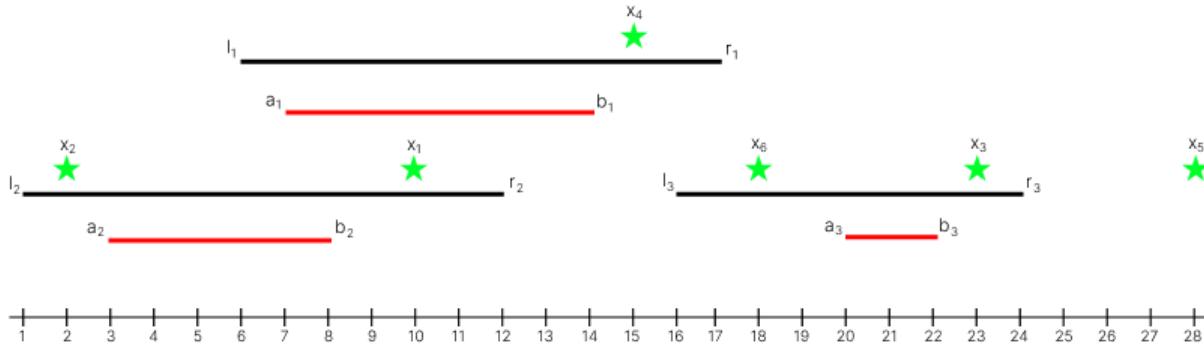
```

14 14 23 15 28 22
14 14 14 14 22 23 14 14 15
7 8 7
15 14 14 30 24 17 31 4 8
32 32 32 5 8 33 4 32

```

Примечание

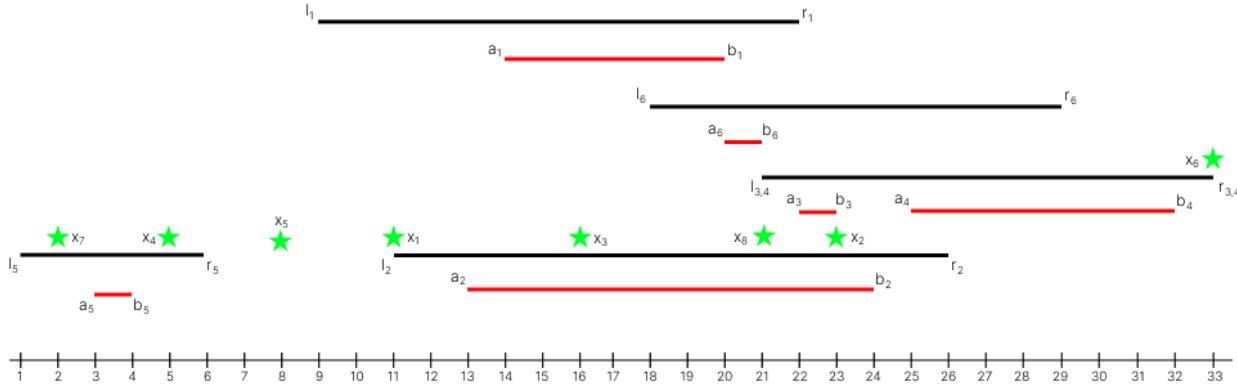
В первом наборе входных данных:



Оптимальные действия при старте из каждой позиции:

1. Воспользуемся порталом 1 и телепортируемся в точку $b_1 = 14$.
2. Воспользуемся сначала порталом 2 и телепортируемся в точку 6, находящуюся на отрезке $[l_1, r_1] = [6, 17]$, после чего воспользуемся порталом 1 и телепортируемся в точку $b_1 = 14$.
3. Останемся в точке $x_3 = 23$, не пользуясь порталами.
4. Останемся в точке $x_4 = 15$, не пользуясь порталами.
5. Точка $x_5 = 28$ не находится ни на одном отрезке, поэтому Андрей не сможет никуда телепортироваться.
6. Точка $x_6 = 18$ находится на отрезке $[l_3, r_3] = [16, 24]$, воспользуемся порталом 3 и телепортируемся в точку $b_3 = 22$.

В пятом наборе входных данных:



Оптимальные действия при старте из каждой позиции:

1. Воспользуемся сначала порталом 1 и телепортируемся в точку 15 на отрезке $[a_1, b_1] = [14, 20]$, после чего воспользуемся порталом 2 и телепортируемся в точку 21, находящуюся на отрезке $[l_4, r_4] = [21, 33]$ и на отрезке $[a_2, b_2] = [13, 24]$, после телепортируемся в точку $b_4 = 32$.
2. Воспользуемся сначала порталом 6 и телепортируемся в точку 20 на отрезке $[a_6, b_6] = [20, 21]$, после чего воспользуемся порталом 2 и телепортируемся в точку 22, находящуюся одновременно на отрезке $[l_4, r_4] = [21, 33]$ и на отрезке $[a_2, b_2] = [13, 24]$, после телепортируемся в точку $b_4 = 32$.
3. Сделаем те же самые действия что из первой позиции.
4. Останемся в точке $x_4 = 5$, не пользуясь порталами.
5. Точка 8 не находится ни на одном отрезке, поэтому Андрей не сможет никуда телепортироваться.
6. Останемся в точке $x_6 = 33$, не пользуясь порталами.
7. Воспользуемся порталом 5 и телепортируемся в точку $b_5 = 4$.
8. Сделаем те же самые действия что и из первой позиции.



D. Сортировка умножением

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам задан массив a длины n , состоящий из **положительных целых чисел**.

Вы можете применять следующую операцию любое количество раз (возможно, ноль раз):

- выбрать три целых числа l , r и x , для которых $1 \leq l \leq r \leq n$, и умножить каждое a_i , для которого $l \leq i \leq r$, на x .

Обратите внимание, что вы можете выбрать **любое** целое число x , не обязательно положительное.

Вам нужно определить минимальное число таких операций, за которое вы можете сделать массив a отсортированным **строго в порядке возрастания** (т. е. должно выполняться условие $a_1 < a_2 < \dots < a_n$).

Входные данные

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

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

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

Дополнительное ограничение на входные данные: сумма всех n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно число — минимальное количество операций, необходимых для превращения массива a в отсортированный в порядке возрастания массив.

Пример

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

Примечание

В первом наборе входных данных мы можем выполнять следующие операции:

- $l = 2, r = 4, x = 3$;
- $l = 4, r = 4, x = 2$;
- $l = 5, r = 5, x = 10$.

После этих операций массив a превратится в $[1, 3, 6, 12, 20]$.

Во втором наборе входных данных мы можем выполнять следующие операции:

- $l = 1, r = 4, x = -2$;
- $l = 6, r = 6, x = 1337$.

После этих операций массив a превратится в $[-10, -8, -6, -4, 5, 1337]$.

В третьем наборе входных данных массив a уже отсортирован.


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

D. Корова и поля

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Бесси пасется на ферме, состоящей из n полей, соединенных с помощью m двусторонних дорог. Сейчас она находится на поле 1, и вернется домой на поле n в конце дня.

Кор-федерация Коровников приказала Фермеру Джону построить одну дополнительную двустороннюю дорогу. У фермы есть k особых полей, и было решено построить дорогу между двумя различными особыми полями. Фермер Джон может построить дорогу даже между полями, которые уже соединены дорогой.

Уже после постройки дороги, Бесси вернется по кратчайшему пути от поля 1 к полю n . Так как Бесси нужно больше тренироваться, Фермер Джон должен **максимизировать** длину данного кратчайшего пути. Помогите ему!

Входные данные

В первой строке заданы три целых числа n , m и k ($2 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 2 \cdot 10^5$, $2 \leq k \leq n$) — количество полей на ферме, количество дорог и количество особых полей.

Во второй строке заданы k целых чисел a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$) — номера особых полей. Все a_i различны.

В i -й из следующих m строк задано два целых числа x_i и y_i ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$), описывающих двустороннюю дорогу между полями x_i и y_i .

Гарантируется, что из любого поля можно добраться до любого другого поля. Также гарантируется, что для любой пары полей существует не более одной дороги, непосредственно соединяющей их.

Выходные данные

Выведите единственное число — **максимально** возможную длину кратчайшего пути из 1 в n после того, как Фермер Джон построит одну дорогу оптимально.

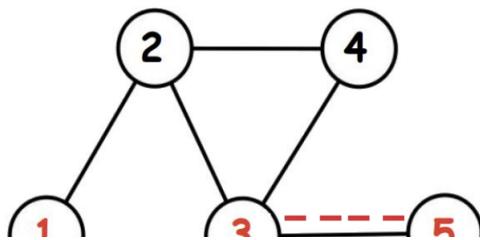
Примеры

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

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

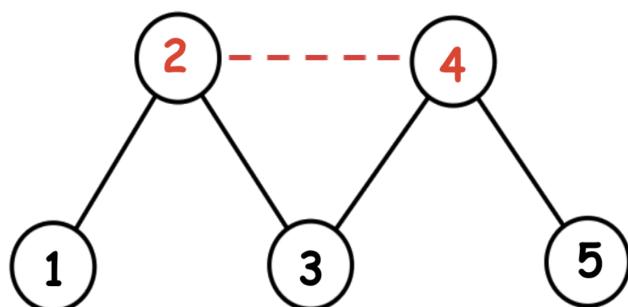
Примечание

Граф из первого примера изображен ниже. Особые поля отмечены красным. Для Фермера Джона оптимально построить дорогу между полями 3 и 5, тогда кратчайший путь из 1 в 5 будет равен 3.





Граф из второго примера изображен ниже. Фермер Джон должен построить дорогу между полями 2 и 4, тогда кратчайший путь из 1 в 5 будет равен 3.



[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 18.11.2025 11:18:37_{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



| ИТМО


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

G. Дерево минимального ора

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

В последнее время Влад увлёкся остовными деревьями, так что его друзья не долго думая подарили ему на день рождения связный взвешенный неориентированный граф из n вершин и m рёбер.

Влад определил орность остовного дерева как побитовое ИЛИ всех его весов и теперь его интересует, какова минимальная возможная орность, которой можно добиться, выбрав некоторое остовное дерево. Остовное дерево — связный подграф данного графа, не содержащий циклов.

Иными словами вы хотите оставить $n - 1$ ребро, так чтобы граф остался связным и побитовое ИЛИ весов рёбер было как можно меньше. Вы должны найти само побитовое ИЛИ.

Входные данные

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

Перед каждым набором в тесте записана пустая строка.

Далее следуют два числа n и m ($3 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 2 \cdot 10^5$) — количество вершин и рёбер графа, соответственно.

Следующие m строк содержат описание рёбер. Стока i содержит три числа v_i , u_i и w_i ($1 \leq v_i, u_i \leq n$, $1 \leq w_i \leq 10^9$, $v_i \neq u_i$) — вершины, которые соединяет ребро, и его вес.

Гарантируется, что сумма m и сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$ и каждый тест содержит связный граф.

Выходные данные

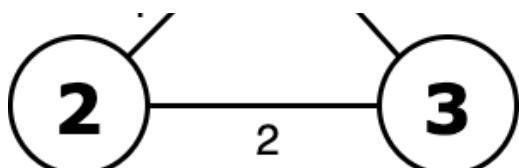
Выведите t строк, каждая из которых содержит ответ на соответствующий набор входных данных — минимальную возможную орность остовного дерева.

Пример

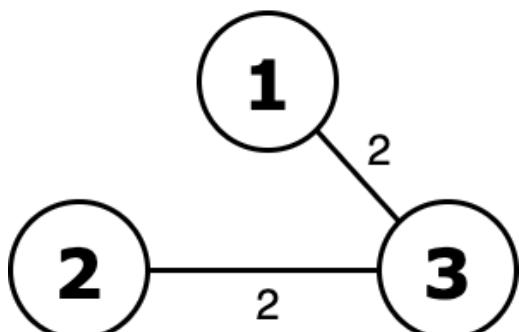
входные данные	Скопировать
3	
3 3	
1 2 1	
2 3 2	
1 3 2	
5 7	
4 2 7	
2 5 8	
3 4 2	
3 2 1	
2 4 2	
4 1 2	
1 2 2	
3 4	
1 2 1	
2 3 2	
1 3 3	
3 1 4	
выходные данные	Скопировать
2	
10	
3	

Примечание

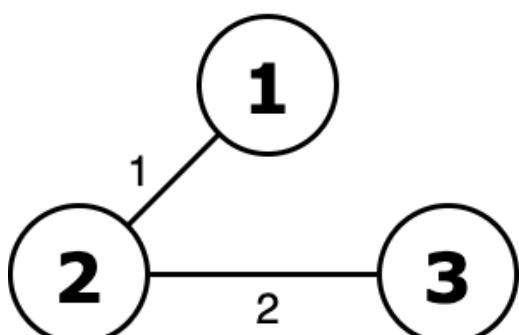




Изначальный граф из первого примера.



Ориентированность такого дерева равна $2 \text{ or } 2 = 2$ и является оптимальной.



Оставив ребро веса 1 мы получим ориентированность $1 \text{ or } 2 = 3$.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 18.11.2025 11:18:37 UTC+5 (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



| ИТМО

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

C2. Небоскрёбы (усложнённая версия)

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 512 мегабайт

Это более сложная версия задачи. В этой версии $n \leq 500\,000$

В Берляндии активно застраивается окраина столицы. Компания «Kernel Panic» руководит постройкой жилого комплекса из небоскрёбов в Новой Берлске. Все небоскрёбы строятся вдоль шоссе. Известно, что компания уже купила n участков возле шоссе и готовится возвести n небоскрёбов, по одному зданию на один участок.

Архитекторы при планировании зданий должны учитывать несколько требований. Во-первых, поскольку земля на каждом участке имеет разные свойства, для каждого небоскрёба есть свое ограничение по количеству этажей, которое он может иметь. Во-вторых, согласно дизайн-коду города, недопустима ситуация, когда для какого-то небоскрёба сразу по обе стороны от него есть небоскрёбы выше него.

Более формально, пронумеруем участки целыми числами от 1 до n . Тогда у небоскрёба на участке с номером i количество этажей a_i не может быть больше m_i ($1 \leq a_i \leq m_i$). Также не может быть, что на плане существуют два участка с номерами j и k , таких что $j < i < k$ и $a_j > a_i < a_k$. Участки j и k не обязаны быть соседними с i .

Компания хочет, чтобы суммарное количество этажей в построенных небоскрёбах было как можно больше. Помогите ей выбрать количество этажей для каждого небоскрёба оптимальным образом, то есть так, чтобы выполнялись все ограничения, а среди всех таких вариантов выберите один из планов, в котором суммарное количество этажей максимально возможно.

Входные данные

В первой строке задано одно целое число n ($1 \leq n \leq 500\,000$) — количество участков.

Вторая строка содержит целые числа m_1, m_2, \dots, m_n ($1 \leq m_i \leq 10^9$) — максимально возможное количество этажей для небоскрёба на каждом участке.

Выходные данные

Выведите n чисел a_i — количества этажей в плане для каждого небоскрёба, такие, что выполняются все ограничения, а суммарное количество этажей во всех небоскрёбах максимально возможное.

Если возможно несколько ответов, выведите любой.

Примеры

входные данные	Скопировать
5	
1 2 3 2 1	
выходные данные	Скопировать
1 2 3 2 1	
входные данные	Скопировать
3	
10 6 8	
выходные данные	Скопировать
10 6 6	

Примечание

В первом примере можно построить все небоскрёбы с максимально возможной высотой.

Во втором примере придать максимальную высоту всем небоскрёбам нельзя, так как это нарушает ограничение дизайн-кода. Ответ $[10, 6, 6]$ является оптимальным. Обратите внимание, что ответ $[6, 6, 8]$ также удовлетворяет всем ограничениям, но оптимальным не является.


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

D. Смешная игра

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

У Вани есть граф из n вершин (пронумерованных от 1 до n) и массив a из n целых чисел, изначально в графе нет рёбер. Ване стало скучно, и чтобы ему стало весело, он решил выполнить $n - 1$ операцию.

Операция под номером x (операции нумеруются по порядку начиная с 1) заключается в следующем:

- Выбрать 2 различных числа $1 \leq u, v \leq n$, таких что $|a_u - a_v|$ делится на x .
- Добавить в граф неориентированное ребро между вершинами u и v .

Помогите Ване с помощью $n - 1$ операции получить связный* график, или скажите, что это невозможно.

*Граф называется связным, если в нём можно от любой вершины дойти до любой другой, двигаясь по рёбрам.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных. Далее следует описание наборов входных данных.

В первой строке каждого набора входных данных дано число n ($1 \leq n \leq 2000$) — количество вершин в графе.

Во второй строке каждого набора входных данных дано n 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 2000.

Выходные данные

Для каждого набора входных данных, если решения не существует, то выведите «No» (без кавычек).

Иначе выведите «Yes» (без кавычек), после этого выведите $n - 1$ строку, в i -й из них выведите числа u и v , которые надо выбрать на операции i .

Вы можете вывести каждую букву в любом регистре (например, «YES», «Yes», «yes», «yEs» будут распознаны как положительный ответ).

Пример

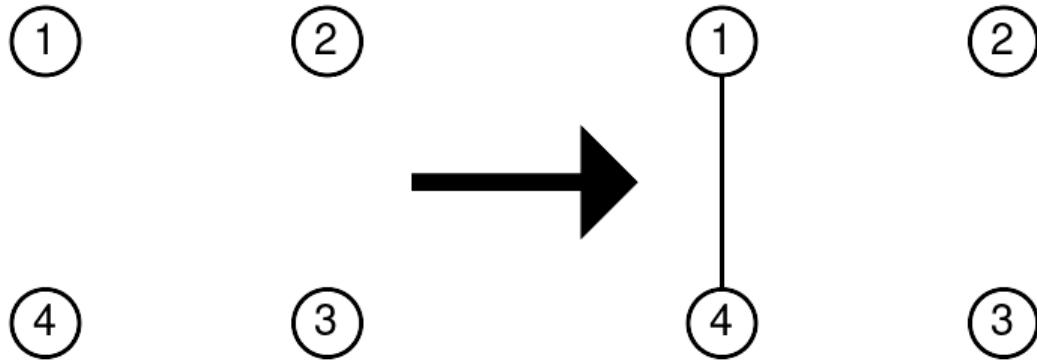
входные данные	Скопировать
8	
2	
1 4	
4	
99 7 1 13	
5	
10 2 31 44 73	
5	
87 6 81 44 32	
5	
62 35 33 79 16	
5	
6 51 31 69 42	
5	
52 63 25 21 5	
12	
33 40 3 11 31 43 37 8 50 5 12 22	
выходные данные	Скопировать
YES	
2 1	
YES	
4 1	
2 1	
3 2	
YES	
5 1	
4 1	
3 1	
2 1	
YES	
4 1	
3 1	
2 1	

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

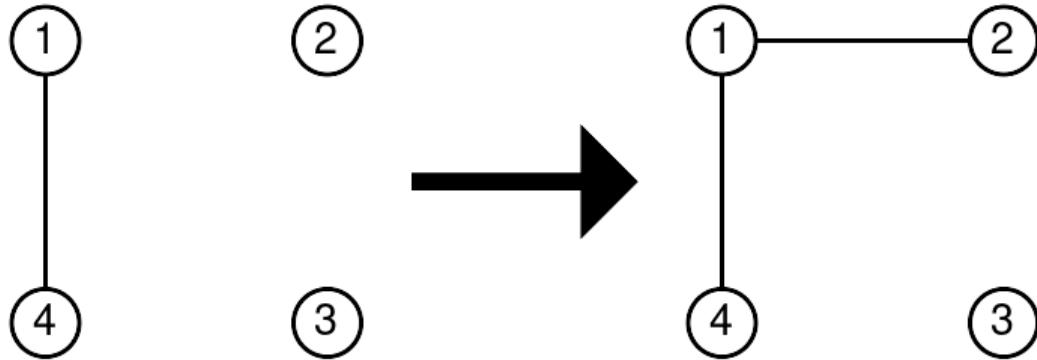
Примечание

Рассмотрим второй набор входных данных.

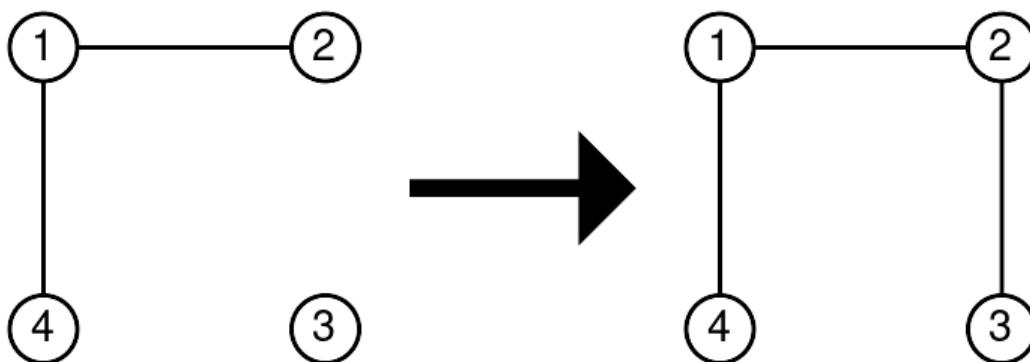
- Первая операция ($x = 1$): можно соединить вершины 4 и 1, так как $|a_4 - a_1| = |13 - 99| = |-86| = 86$, а 86 кратно 1.



- Вторая операция ($x = 2$): можно соединить вершины 2 и 1, так как $|a_2 - a_1| = |7 - 99| = |-92| = 92$, а 92 кратно 2.



- Третья операция ($x = 3$): можно соединить вершины 3 и 2, так как $|a_3 - a_2| = |1 - 7| = |-6| = 6$, а 6 кратно 3.



Из картинки видно, что получился связный граф.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 18.11.2025 11:18:38^{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке




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

F. Ценные карточки

ограничение по времени на тест: 4 секунды

ограничение по памяти на тест: 512 мегабайт

В своём любимом кафе Kmes захотел в очередной раз отведать селёдку под шубой. Раньше ему бы не доставило труда сделать это, однако кафе недавно выдвинуло новую политику покупки.

Теперь для её совершения Kmes надо решить следующую задачу: перед ним на стол выкладывают n карточек с ценами на разные позиции, на i -й карточке написано число a_i , среди этих цен нет целого положительного числа x .

Kmes просят разбить эти карточки на минимальное количество плохих отрезков (так, чтобы каждая карточка принадлежала ровно одному отрезку). Плохим считается такой отрезок, что в нём нельзя выбрать подмножество карточек с произведением, равным x . Все отрезки, на которые Kmes разобьёт карточки, должны быть плохими.

Формально, отрезок (l, r) плохой, если не существует индексов $i_1 < i_2 < \dots < i_k$, что $l \leq i_1, i_k \leq r$, и $a_{i_1} \cdot a_{i_2} \dots \cdot a_{i_k} = x$. Помогите Kmes определить минимальное количество плохих отрезков, чтобы насладиться любимым блюдом.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^3$) — количество наборов тестовых данных.

В первой строке каждого набора входных данных вам даны 2 числа n и x ($1 \leq n \leq 10^5$, $2 \leq x \leq 10^5$) — количество карточек и число соответственно.

Во второй строке каждого набора входных данных содержится n чисел a_i ($1 \leq a_i \leq 2 \cdot 10^5$, $a_i \neq x$) — цены на карточках.

Гарантируется, что сумма n по всем наборам тестовых данных не превышает 10^5 .

Выходные данные

Для каждого набора входных данных выведите минимальное количество плохих отрезков.

Пример

входные данные	Скопировать
<pre>8 6 4 2 3 6 2 1 2 9 100000 50000 25000 12500 6250 3125 2 4 8 16 5 2 1 1 1 1 1 8 6 4 3 4 3 4 3 4 3 7 12 6 11 1 3 11 10 2 10 5 2 4 4 2 4 4 4 3 1 1 7 8 4 6 5 1 2 4 1 8 27 3 9 17 26 2 20 9 3</pre>	
выходные данные	Скопировать
<pre>3 2 1 1 2 1 3 3</pre>	


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

C. Выполните операции и максимизируйте счёт

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Я вижу хендл *satyam343*. У меня идут мурашки по телу. Пожалуйста, в этот раз больше задач про медианы. Я их обожаю. Пожалуйста, *satyam343*, мы верим в тебя.

 — Самый большой поклонник *satyam343*

Вам дан массив a длины n и целое число k . Вам также дан бинарный массив b длины n .

Вы можете выполнить следующую операцию не более k раз:

- Выбрать индекс i ($1 \leq i \leq n$) такой, что $b_i = 1$. Установить $a_i = a_i + 1$ (т.е. увеличить a_i на 1).

Ваш счёт определяется как $\max_{i=1}^n (a_i + \text{median}(c_i))$, где c_i обозначает массив длины $n - 1$, который вы получаете, удаляя a_i из a . Другими словами, ваш счёт — это максимальное значение $a_i + \text{median}(c_i)$ среди всех i от 1 до n .

Найдите максимальный счёт, который можно получить при оптимальном выполнении операций.

Для произвольного массива p , $\text{median}(p)$ определяется как $\left\lfloor \frac{|p|+1}{2} \right\rfloor$ -й элемент в отсортированном p . Например, $\text{median}([3, 2, 1, 3]) = 2$ и $\text{median}([6, 2, 4, 5, 1]) = 4$.

Входные данные

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

Каждый набор входных данных начинается с двух целых чисел n и k ($2 \leq n \leq 2 \cdot 10^5$, $0 \leq k \leq 10^9$) — длина a и количество операций, которые вы можете выполнить.

Следующая строка содержит n разделенных пробелом целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — массив a .

Следующая строка содержит n разделенных пробелом целых чисел b_1, b_2, \dots, b_n (b_i равно 0 или 1) — массив b .

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

Выходные данные

Для каждого набора входных данных выведите на отдельной строке максимальное значение счёт, которое вы можете получить.

Пример

входные данные	Скопировать
<pre>8 2 10 3 3 1 1 3 10 3 3 3 0 0 0 4 4 2 1 5 1 0 1 0 1 5 4 7 5 2 5 4 0 0 1 0 1 5 1 5 15 15 2 11 1 0 0 1 1 5 2 10 11 4 10 15 1 1 0 1 0 4 4 1 1 2 5 1 1 0 0 2 1000000000 1000000000 1000000000 1 1</pre>	

выходные данные**Скопировать**

```
16
6
8
13
21
26
8
3000000000
```

Примечание

Для первого набора входных данных оптимально выполнить 5 операций над обоими элементами и получить $a = [8, 8]$. Таким образом, максимальная оценка, которую мы можем получить, равна $\max(8 + \text{median}[8], 8 + \text{median}[8]) = 16$, так как $c_1 = [a_2] = [8]$. Можно доказать, что получить больший счёт невозможно.

Для второго набора входных данных вы не можете выполнить операции ни над одним элементом, поэтому a остается $[3, 3, 3]$. Таким образом, максимальная оценка, которую мы можем получить, равна $3 + \text{median}[3, 3] = 6$, так как $c_1 = [a_2, a_3] = [3, 3]$. Можно доказать, что получить больший счёт невозможно.

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

Время на сервере: 18.11.2025 11:18:42^{UTC+5} (h1).

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

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

При поддержке



ИТМО

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Разделение массива

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам заданы массив a_1, a_2, \dots, a_n и число k .

Вам нужно разделить массив на k непустых последовательных подотрезков. Каждый элемент в массиве должен быть покрыт ровно одним подотрезком. Пусть $f(i)$ равно индексу подотрезка, которому принадлежит i -й элемент. Подотрезки нумеруются слева направо и от 1 до k .

Тогда цена конкретного разбиения будет равна $\sum_{i=1}^n (a_i \cdot f(i))$. Например, если $a = [1, -2, -3, 4, -5, 6, -7]$, и мы разделили его на 3 подотрезка следующим способом: $[1, -2, -3], [4, -5], [6, -7]$, тогда цена разбиения будет равна $1 \cdot 1 - 2 \cdot 1 - 3 \cdot 1 + 4 \cdot 2 - 5 \cdot 2 + 6 \cdot 3 - 7 \cdot 3 = -9$.

Посчитайте максимальную цену, которую вы можете получить, разделив массив a на k непустых последовательных подотрезков.

Входные данные

Первая строка содержит два числа n и k ($1 \leq k \leq n \leq 3 \cdot 10^5$).

Вторая строка содержит n чисел a_1, a_2, \dots, a_n ($|a_i| \leq 10^6$).

Выходные данные

Выведите максимальную цену, которую вы можете получить, разделив массив a на k непустых последовательных подотрезков.

Примеры

входные данные	выходные данные
5 2 -1 -2 5 -4 8	15
входные данные	выходные данные
7 6 -3 0 -1 -2 -2 -4 -1	-45
входные данные	выходные данные
4 1 3 -1 6 0	8




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

D. Кузнечное дело

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Вы играете в одну известную игру (которая «просто работает»), в которой у вас есть различные навыки, которые можно прокачивать. Сегодня вы сосредоточились на навыке «Кузнечное дело». Ваша тактика довольно проста: ковать оружие из слитков, а затем их переплавлять, чтобы вернуть часть материалов. Для простоты, каждый раз, когда вы создаете предмет, вы получаете 1 очко опыта, и каждый раз, когда вы переплавляете предмет, вы также получаете 1 очко опыта.

В игре есть n классов оружия, которые вы можете создать, и m типов металлических слитков.

Вы можете создать одно оружие i -го класса, затратив a_i слитков металла *одного типа*. Переплавка одного оружия i -го класса (которое вы ранее создали) возвращает вам b_i слитков металла, *из которого оно было изготовлено*.

У вас есть c_j слитков металла j -го типа, и вы знаете, что можете создать оружие любого класса из любого типа металла. Каждую комбинацию класса оружия и типа металла можно использовать любое количество раз.

Какое максимальное количество очков опыта вы можете заработать, создавая и переплавляя оружие?

Входные данные

В первой строке заданы два целых числа n и m ($1 \leq n, m \leq 10^6$) — количество классов оружия и типов металла.

Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$), где a_i — это количество слитков, необходимых для создания одного оружия i -го класса.

В третьей строке заданы n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i < a_i$), где b_i — количество слитков, возвращаемых при переплавке одного оружия i -го класса, которое вы ранее создали.

В четвертой строке заданы m целых чисел c_1, c_2, \dots, c_m ($1 \leq c_j \leq 10^9$) — количество слитков металла соответствующего типа, которые у вас есть.

Выходные данные

Выведите одно целое число — максимальное общее количество очков опыта, которое вы можете заработать, многократно создавая и переплавляя оружие.

Примеры

входные данные	Скопировать
5 3 9 6 7 5 5 8 4 5 1 2 10 4 7	
выходные данные	Скопировать
12	
входные данные	Скопировать
3 4 10 20 20 0 0 0 9 10 19 20	
выходные данные	Скопировать
8	
входные данные	Скопировать
1 5 3 1 1000000000 1000000000 1000000000 1000000000 1000000000	
выходные данные	Скопировать
4999999990	

Примечание

В первом teste вы можете сделать следующее:

1. создать одно оружие 1-го класса из 1-го типа металла, потратив 9 слитков;
2. переплавить это оружие, вернув 8 слитков 1-го типа металла;
3. снова создать и переплавить одно оружие 1-го класса из 1-го типа металла;
4. создать и переплавить одно оружие 3-го класса из 1-го типа металла;
5. создать и переплавить одно оружие 3-го класса из 3-го типа металла;
6. создать и переплавить одно оружие 4-го класса из 1-го типа металла;
7. создать и переплавить одно оружие 5-го класса из 3-го типа металла;

В конце у вас останется $c = [2, 4, 2]$ слитков. Всего вы создали 6 оружий и переплавили 6 оружий, заработав в общей сложности 12 очков опыта.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 18.11.2025 11:18:46_{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке




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

D. Шлюзы

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Недавно в Диваново построили огромную шлюзовую систему. Всего было построено n шлюзов, i -й из них имеет объем v_i литров, то есть в любой момент времени в нем может находиться любой объем воды от 0 до v_i литров. В каждый шлюз ведет труба, при открытии которой в шлюз будет поступать по 1 литру воды в секунду.

Шлюзовая система устроена так, что если доливать воду в i -й шлюз сверх его объема, то он будет моментально передавать всю лишнюю воду шлюзу с номером $i + 1$. Если шлюз с номером $i + 1$ тоже станет заполнен, то вся лишняя вода будет переливаться дальше. Вода из последнего шлюза будет выливаться в реку.

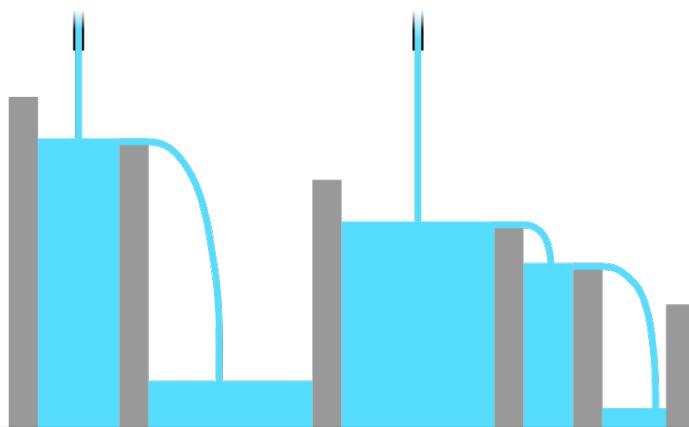


Рисунок показывает 5 шлюзов с открытыми трубами к шлюзам 1 и 3. Так как шлюзы 1, 3 и 4 уже заполнены, фактически вода идет в шлюзы 2 и 5. Обратите внимание, что объем i -го шлюза может быть больше объема $i + 1$ -го шлюза.

Для того, чтобы шлюзы начали функционировать, необходимо заполнить каждый из них. Мэра Дивановской области интересует q независимых запросов. Для каждого запроса предположим, что изначально все шлюзы пусты и все трубы закрыты, затем одновременно открываются несколько труб. Для j -го запроса мэр хочет знать, какое минимальное число труб надо включить, чтобы не позже чем через t_j секунд все шлюзы стали заполнены.

Помогите мэру справиться с этой сложной задачей, и ответьте на все его запросы!

Входные данные

В первой строке вводится одно целое число n ($1 \leq n \leq 200\,000$) — количество шлюзов.

Во второй строке вводятся n целых чисел v_1, v_2, \dots, v_n ($1 \leq v_i \leq 10^9$) — объемы шлюзов.

В третьей строке вводится одно целое число q ($1 \leq q \leq 200\,000$) — число запросов.

В следующих q строках вводится по одному целому числу t_j ($1 \leq t_j \leq 10^9$) — время, за которое нужно наполнить все шлюзы в j -м запросе.

Выходные данные

Выведите q чисел. j -е из них должно быть равно минимальному числу труб, которые придётся открыть, чтобы наполнить все шлюзы за время t_j . Если за это время наполнить всю шлюзы невозможно, то выведите -1 .

Примеры

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

```
-1  
4  
3
```

входные данные**Скопировать**

```
5  
4 4 4 4 4  
6  
1  
3  
6  
5  
2  
4
```

выходные данные**Скопировать**

```
-1  
-1  
4  
4  
-1  
5
```

Примечание

В первом примере есть 6 запросов.

В запросах 1, 3, 4 ответ —1. Чтобы заполнить первый шлюз нужно подождать 4 секунды, даже если открыть все трубы.

Во шестом запросе можно открыть трубы в шлюзах 1, 3 и 4. Так через 4 секунды заполняются шлюзы 1 и 4. В следующую 1 секунду 1 литр воды перельётся в шлюзы 2 и 5. Шлюз 3 будет заполнен своей трубой.

Аналогично во втором запросе можно открыть трубы в шлюзах 1, 3 и 4.

В пятом запросе можно открыть трубы в шлюзах с номерами 1, 2, 3, 4.

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

Соревнования по программированию 2.0

Время на сервере: 18.11.2025 11:18:48^{UTC+5} (h1).

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

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

При поддержке



ИТМО


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

D. XOR на дереве

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 512 мегабайт

Дано дерево с n вершинами, пронумерованными от 1 до n . В вершине i записано целое число a_i для каждого $i = 1, 2, \dots, n$. Ваша цель — сделать все a_i одинаковыми, выполнив несколько заклинаний (возможно, ни одного).

Предположим, что корнем дерева выбрана некоторая вершина. Чтобы выполнить заклинание, можно выбрать произвольную вершину v и произвольное неотрицательное целое число c . Затем для каждой вершины i в поддереве[†] v нужно заменить a_i на $a_i \oplus c$. Стоимость такого заклинания равна $s \cdot c$, где s — число вершин в поддереве. Здесь \oplus обозначает операцию **побитового исключающего ИЛИ**.

Пусть m_r — минимально возможная суммарная стоимость, за которую можно сделать все a_i равными, в случае если вершина r выбрана корнем дерева. Найдите m_1, m_2, \dots, m_n .

[†] Пусть вершина r выбрана корнем дерева. Тогда вершина i принадлежит поддереву v , если простой путь от i до r содержит v .

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.

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

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i < 2^{20}$).

Каждая из следующих $n - 1$ строк содержит два целых числа u и v ($1 \leq u, v \leq n$), означающих наличие ребра между вершинами u и v .

Гарантируется, что заданные ребра образуют дерево.

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

Выходные данные

Для каждого набора входных данных выведите m_1, m_2, \dots, m_n на отдельной строке.

Пример

входные данные	Скопировать
2	
4	
3 2 1 0	
1 2	
2 3	
2 4	
1	
100	
выходные данные	Скопировать
8 6 12 10	
0	

Примечание

В первом наборе входных данных, чтобы найти m_1 , подвесим дерево за вершину 1.

- На первом заклинании выберем $v = 2$ и $c = 1$. После выполнения этого заклинания a станет равным $[3, 3, 0, 1]$. Стоимость этого заклинания равна 3.
- На втором заклинании выберем $v = 3$ и $c = 3$. После выполнения этого заклинания a станет равным $[3, 3, 3, 1]$. Стоимость этого заклинания равна 3.
- На третьем заклинании выберем $v = 4$ и $c = 2$. После выполнения этого заклинания a станет равным $[3, 3, 3, 3]$. Стоимость этого заклинания равна 2.

Теперь все значения в массиве a одинаковы, а суммарная стоимость равна $3 + 3 + 2 = 8$.

Значения m_2, m_3, m_4 находятся аналогично.

Во втором наборе входных данных цель уже достигнута, поскольку в дереве всего одна вершина.

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

C. Три сумки

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Даны **три** сумки. Каждая содержит непустое мультимножество чисел. Над сумками можно производить ряд операций. В ходе каждой операции разрешается выбрать любые две непустые сумки и по одному числу из каждой из этих сумок. Допустим, из первой сумки было выбрано число a , а из второй — число b . После этого число b убирается из второй сумки, а число a в первой сумке заменяется на $a - b$. Обратите внимание, что если эти числа встречаются несколько раз, то убирается/заменяется лишь одно вхождение.

Все операции должны производиться таким образом, чтобы в итоге осталось лишь одно число в одной из сумок (две другие должны быть пустыми). Можно показать, что всегда существует такая комбинация операций, которая приведет к необходимой конечной конфигурации. Среди всех этих конфигураций требуется найти ту, при которой в конце останется наибольшее число.

Входные данные

Первая строка входных данных содержит три целых числа n_1, n_2 и n_3 ($1 \leq n_1, n_2, n_3 \leq 3 \cdot 10^5$, $1 \leq n_1 + n_2 + n_3 \leq 3 \cdot 10^5$), разделенных пробелами — количество чисел в трех сумках.

Каждая i -я из трех последующих строк содержит n_i целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,n_i}$ ($1 \leq a_{i,j} \leq 10^9$), разделенных пробелами — числа в мультимножестве i -й сумки.

Выходные данные

Выведите единственное целое число — максимальное из тех, что могут быть получены в конечной конфигурации.

Примеры

входные данные	Скопировать
2 4 1 1 2 6 3 4 5 5	
выходные данные	Скопировать
20	
входные данные	Скопировать
3 2 2 7 5 4 2 9 7 1	
выходные данные	Скопировать
29	

Примечание

Для первого примера из условия произведем следующие операции:

[1, 2], [6, 3, 4, 5], [5]

[-5, 2], [3, 4, 5], [5] (Применяем операцию к (1, 6))

[-10, 2], [3, 4], [5] (Применяем операцию к (-5, 5))

[2], [3, 4], [15] (Применяем операцию к (5, -10))

[-1], [4], [15] (Применяем операцию к (2, 3))

[-5], [], [15] (Применяем операцию к (-1, 4))

[], [], [20] (Применяем операцию к (15, -5))

Легко удостовериться в том, что большее число получить нельзя. Таким образом, ответом будет 20.


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

F. Образование

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Поликарп мечтает о покупке нового компьютера, который стоит c тугриков. Для этого он хочет устроиться программистом в большую компанию.

В компании Поликарпа есть n должностей, пронумерованных начиная с единицы. Сотрудник на должности i каждый день зарабатывает $a[i]$ тугриков. Чем выше номер должности, тем больше тугриков получает сотрудник. Изначально Поликарп устраивается на должность с номером 1 и имеет 0 тугриков.

Каждый день Поликарп может сделать одно из двух действий:

- Если Поликарп находится на должности x , то он может заработать $a[x]$ тугриков;
- Если Поликарп находится на должности x ($x < n$) и имеет хотя бы $b[x]$ тугриков, то он может потратить $b[x]$ тугриков на онлайн-курс и перейти на должность $x + 1$.

Например, если $n = 4$, $c = 15$, $a = [1, 3, 10, 11]$, $b = [1, 2, 7]$, то Поликарп может действовать так:

- В первый день Поликарп находится на должности 1 и зарабатывает 1 тугрик. Теперь у него есть 1 тугрик;
- Во второй день Поликарп находится на должности 1 и переходит на должность 2. Теперь у него есть 0 тугриков;
- В третий день Поликарп находится на должности 2 и зарабатывает 3 тугрика. Теперь у него есть 3 тугрика;
- В четвертый день Поликарп находится на должности 2 и переходит на должность 3. Теперь у него есть 1 тугрик;
- В пятый день Поликарп находится на должности 3 и зарабатывает 10 тугриков. Теперь у него есть 11 тугриков;
- В шестой день Поликарп находится на должности 3 и зарабатывает 10 тугриков. Теперь у него есть 21 тугриков;
- Спустя шесть дней, Поликарп может купить себе новый компьютер.

Найдите минимальное количество дней, через которое Поликарп сможет купить себе новый компьютер.

Входные данные

В первой строке содержится одно целое число t ($1 \leq t \leq 10^4$). Далее следуют t наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и c ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq c \leq 10^9$) — количество должностей в компании и стоимость нового компьютера.

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

Третья строка каждого набора входных данных содержит $n - 1$ целое число b_1, b_2, \dots, b_{n-1} ($1 \leq b_i \leq 10^9$).

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

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальное количество дней, через которое Поликарп сможет купить себе новый компьютер.

Пример

входные данные	Скопировать
<pre>3 4 15 1 3 10 11 1 2 7 4 100 1 5 10 50 3 14 12 2 1000000000 1 1 1</pre>	
выходные данные	Скопировать
<pre>6 13 1000000000</pre>	

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

F. Закрась строки и столбцы

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

У вас есть n прямоугольников, i -й из которых имеет ширину a_i и высоту b_i .

Вы можете неограниченное количество раз выполнить следующую операцию: выбрать прямоугольник и клетку в нём, а затем закрасить её.

Каждый раз, когда вы закрашиваете целиком какую-либо строку или столбец, вы получаете 1 балл. Ваша задача — набрать хотя бы k баллов за как можно меньшее число операций.

Допустим, у вас есть прямоугольник ширины 6 и высоты 3. Вы можете набрать 4 очка, закрасив все клетки в 4-х любых столбцах, выполнив таким образом 12 операций.

Входные данные

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

Первая строка описания каждого набора входных данных содержит два целых числа n и k ($1 \leq n \leq 1000$, $1 \leq k \leq 100$) — количество прямоугольников в наборе и требуемое количество баллов.

Следующие n строк содержат описания прямоугольников. i -я строка содержит два целых числа a_i и b_i ($1 \leq a_i, b_i \leq 100$) — ширину и высоту i -го прямоугольника.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит 1000.

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальное количество операций, необходимое для того, чтобы набрать хотя бы k очков. Если набрать хотя бы k очков невозможно, выведите -1.

Пример

входные данные	Скопировать
7 1 4 6 3 1 5 4 4 5 10 1 1 1 1 1 1 1 1 1 1 2 100 1 2 5 6 3 11 2 2 3 3 4 4 3 25 9 2 4 3 8 10 4 18 5 4 8 5 8 3 6 2	
выходные данные	Скопировать
12 14 5 -1 17 80 35	


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

D. Наездник на курице

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Стив по глупости стал копать ночью и наткнулся на монструозное существо: *наездника на курице*ⁿ!

Наездник на курицеⁿ состоит из n мобов, поставленных по порядку друг на друга; моб 1 находится внизу, а моб n — на самом верху. У моба i изначально h_i здоровья.

За одну атаку Стив может нанести 1 урон любому мобу. Если любой моб достигает 0 или менее здоровья, он умирает, и все мобы сверху него падают вниз и формируют новую стопку. Моб, находящийся **внизу** новой стопки, получает 1 урон от падения за каждого моба, который был ниже его до падения (включая того, который только что умер). Это также может его убить, в этом случае все мобы сверху него снова падают вниз, и процесс повторяется.

Например, рассмотрим наездника на курице⁶ с начальными значениями здоровья мобов $[1, 2, 1, 3, 5, 2]$. Если Стив повредит третьего моба в стопке, он умирает, и мобы со здоровьем $[3, 5, 2]$ падают вниз в новую стопку. Нижний моб получает 3 единицы урона от падения, поэтому он также умирает, и мобы со здоровьем $[5, 2]$ падают вниз в новую стопку. Нижний моб получает 1 единицу урона от падения. В итоге, после первой атаки Стива будет две стопки со здоровьем $[1, 2]$ и $[4, 2]$.

Меч Стива не зачарован на починку, поэтому он хочет узнать минимальное количество атак, необходимое для уничтожения всех мобов.

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.

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

Вторая строка каждого набора входных данных содержит n целых чисел h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^9$) — начальное здоровье каждого моба.

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

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальное количество атак, необходимое для уничтожения всех мобов.

Пример

входные данные	Скопировать
5 5 3 1 4 1 2 4 1 1 1 1 6 1 2 1 3 5 2 6 3 1 1 3 2 1 3 1000000000 1000000000 1000000000	
выходные данные	Скопировать
7 1 7 5 299999998	

Примечание

В первом наборе входных данных начальная стопка имеет мобов со здоровьем $[3, 1, 4, 1, 2]$. Стив может использовать одну атаку, чтобы повредить второго моба в стопке и убить его. Третий моб получает 2 единицы урона от падения. Теперь есть две стопки: $[3]$ и $[2, 1, 2]$. Стив может убить второго моба во второй стопке. Третий моб в стопке получает 2 единицы урона от падения, убивая его. Теперь есть две стопки: $[3]$ и $[2]$. Наконец, Стив может использовать пять атак, чтобы убить оставшихся мобов.

Во втором наборе входных данных Стив может нанести 1 урон нижнему мобу в стопке. Когда он умирает, второй моб получает 1 единицу урона от падения и умирает; затем третий моб получает 1 единицу урона от падения и умирает; наконец, четвертый

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

F1. Ближайшее красивое число (упрощённая версия)

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Это упрощённая версия задачи F2. Они отличаются только ограничениями (F1: $k \leq 2$, F2: $k \leq 10$).

Дано число n . Найдите **минимальное** целое число x такое, что $x \geq n$ и x является k -красивым числом.

Число называется k -красивым, если в его десятичной записи, не содержащей лидирующих нулей, встречается **не более k** различных цифр. Например, если $k = 2$, числа 3434443, 55550, 777 и 21 являются k -красивыми, а числа 120, 445435 и 998244353 — нет.

Входные данные

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

Каждый набор входных данных состоит из одной строки, содержащей два целых числа n и k ($1 \leq n \leq 10^9$, $1 \leq k \leq 2$).

Выходные данные

Для каждого набора входных данных в отдельной строке выведите x — минимальное целое k -красивое число, так чтобы $x \geq n$.

Пример

входные данные	Скопировать
4 1 1 221 2 177890 2 998244353 1	
выходные данные	Скопировать
1 221 181111 99999999	

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

Соревнования по программированию 2.0

Время на сервере: 18.11.2025 11:19:01 UTC+5 (h1).

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

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

При поддержке



ITMO

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

A. Бинарная литература

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Бинарная строка — это строка, содержащая только символы 0 и 1.

Коюоті Капоу усердно работает над своей мечтой стать писательницей. Чтобы попрактиковаться, она решила принять участие в Конкурсе по Написанию Бинарных Романов. Указание к конкурсу состоит из трёх бинарных строк длины $2n$. Корректным романом для конкурса является бинарная строка длины не более $3n$, содержащая в качестве подпоследовательностей **не менее двух** из трёх данных строк.

Коюоті только что получила три строки — указание от организаторов конкурса. Помогите ей написать корректный роман для конкурса.

Строка a является подпоследовательностью строки b , если a можно получить из b , удалив несколько (возможно, ноль) символов.

Входные данные

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

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

Каждая из следующих трех строк содержит бинарную строку длины $2n$. Гарантируется, что эти три строки попарно различны.

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

Выходные данные

Для каждого набора входных данных выведите одну строку, содержащую бинарную строку длины не более $3n$, которая в качестве подпоследовательностей содержит как минимум две из заданных бинарных строк.

Можно доказать, что при данных ограничениях такая бинарная строка существует всегда.

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

Пример

входные данные	Скопировать
2	
1	
00	
11	
01	
3	
011001	
111010	
010001	

выходные данные	Скопировать
010	
011001010	

Примечание

В первом наборе входных данных бинарные строки 00 и 01 являются подпоследовательностями выходной строки: **010** и **010**.

Обратите внимание, что 11 не является подпоследовательностью выходной строки, но это не требуется.

Во втором наборе входных данных все три входных строки являются подпоследовательностями выходной строки: **011001010**, **011001010** и **011001010**.

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

D. Ход гения

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

Вам заданы три целых числа a , b и k .

Найдите два двоичных числа x и y ($x \geq y$) такие, что:

1. и x , и y состоят из a нулей и b единиц (в двоичной системе счисления);
2. $x - y$ в двоичной системе содержит ровно k единиц.

В x и y ведущие нули запрещены.

Входные данные

В единственной строке заданы три целых числа a , b и k ($0 \leq a; 1 \leq b; 0 \leq k \leq a + b \leq 2 \cdot 10^5$) — количество нулей, единиц и единиц в результирующем числе.

Выходные данные

В первой строке выведите «Yes», если можно найти два подходящих числа или «No» в противном случае.

В случае, если ответ существует, во второй строке выведите число x в двоичной системе счисления, а в третьей строке выведите число y также в двоичной системе счисления.

Если возможных ответов несколько, то выведите любой.

Примеры

входные данные	Скопировать
4 2 3	
выходные данные	Скопировать
Yes 101000 100001	

входные данные	Скопировать
3 2 1	
выходные данные	Скопировать
Yes 10100 100010	

входные данные	Скопировать
3 2 5	
выходные данные	Скопировать
No	

Примечание

В первом примере $x = 101000_2 = 2^5 + 2^3 = 40_{10}$, $y = 100001_2 = 2^5 + 2^0 = 33_{10}$, $40_{10} - 33_{10} = 7_{10} = 2^2 + 2^1 + 2^0 = 111_2$. Отсюда видно, что в $x - y$ содержатся ровно 3 единицы.

Во втором примере $x = 10100_2 = 2^4 + 2^2 = 20_{10}$, $y = 10010_2 = 2^4 + 2^1 = 18$, $x - y = 20 - 18 = 2_{10} = 10_2$. Ровно одна единица.

В третьем примере из условия можно показать, что ответа нет.