

Задача А. Задача из Магнитогорска

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Жителям Магнитогорска безумно интересно для двух целых чисел a и b узнать, можно ли так переставить цифры числа a , чтобы получилось число **без ведущих нулей**, не большее b ? Ваша задача удовлетворить любопытство магнитогорцев.

Формат входных данных

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.

Единственная строка каждого набора входных данных содержит два целых числа a и b ($1 \leq a, b \leq 999$).

Формат выходных данных

Для каждого набора входных данных выведите «Yes», если можно переставить цифры в числе a , чтобы получилось число без ведущих нулей, не большее b , и «No» иначе.

Вы можете вывести ответ в любом регистре (верхнем или нижнем). Например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительные ответы.

Пример

стандартный ввод	стандартный вывод
9	Yes
1 999	No
999 1	Yes
321 123	Yes
990 989	No
990 908	Yes
1 1	Yes
21 12	Yes
500 501	No
500 499	

Замечание

В первом наборе входных данных число a уже не больше b , поэтому ответ «Yes».

Во втором наборе входных данных как бы мы не переставляли цифры в числе a , оно всегда будет равно 999 и будет больше b .

В третьем наборе входных данных мы можем переставить цифры в числе a и получить число 123, которое не больше b .

В четвёртом наборе входных данных мы можем переставить цифры в числе a и получить число 909, которое не больше b .

Задача В. Задача из Карабаша

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как-то раз группа друзей собралась отметить день рождения. Веселье затянулось до поздней ночи, и в какой-то момент все сильно проголодались. Однако ночью общественный транспорт не ходит, идти до магазина далеко, а такси — дорого. Но кушать хочется!

Самый сообразительный из друзей заметил интересную закономерность: стоимость такси со временем изменяется по рекуррентной формуле, в которой x_0 равно нулю, числа a и b заранее фиксированы и:

$$x_i = a \cdot x_{i-1} + b$$

Например, если $a = 2$ и $b = 1$, то первые несколько стоимостей такси будут следующими: 0, 1, 3, 7, 15, 31.

Друзья — школьники, и у них ограниченный бюджет. Чтобы не попасть в неприятности, они решили действовать осторожно:

- Если цена такси строго больше заданного порога r , они не рискуют и идут в лес собирать грибы.
- Если же цена такси строго меньше другого порога l , они боятся, что за ними может приехать маньяк (ночные страшилки ещё свежи в памяти!), и тоже отправляются в лес.
- Только если цена такси в пределах $[l, r]$ (включая обе границы), друзья считают поездку безопасной и едут в магазин за чипсами.

Ваша задача для t различных сценариев a , b , l и r определить, смогут ли ребята вызвать такси в какой-нибудь момент времени или нет.

Формат входных данных

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

В следующих t строках входных данных описываются сценарии. Каждый сценарий описывается в одной строке в виде четырёх целых чисел a , b , l и r ($1 \leq a, b, l, r \leq 10^9$, $l \leq r$).

Формат выходных данных

Для каждого сценария выведите «Yes», если друзья смогут вызывать такси в какой-нибудь момент времени, и «No» иначе.

Вы можете вывести ответ в любом регистре (верхнем или нижнем). Например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительные ответы.

Пример

стандартный ввод	стандартный вывод
5	Yes
1 2 9 11	No
2 1 16 30	Yes
3 3 31 74	Yes
1 92 920000001 920000092	Yes
1 1 1000000000 1000000000	

Замечание

В первом сценарии первые несколько стоимостей такси будут следующими: 0, 2, 4, 6, 8, 10, 12. В этом случае ребята смогут вызвать такси, когда его стоимость будет равна 10.

Первые несколько стоимостей такси из второго сценария можно увидеть в условии задачи.

Задача С. Задача из Новогорного

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Копатыч с Кар-Карычем переехали в посёлок Новогорный, где Копатыч занялся выводом сорта гигантских тыкв, а Кар-Карыч проводил дни, задумчиво наблюдая за горным пейзажем.

В один из дней Кар-Карыч предложил Копатычу насладиться горами вместе с ним. Для этого он предложил найти Копатычу количество *новогорных* перестановок чисел от 1 до n . Перестановку Кар-Карыч называет *новогорной* в том и только в том случае, если её можно разбить на два **непустых** возрастающих подотрезка чисел. Например, перестановка $(2, 5, 1, 3, 4)$ будет *новогорной*, так как её можно разбить на два возрастающих подотрезка $(2, 5)$ и $(1, 3, 4)$, а перестановка $(3, 2, 1)$ не будет *новогорной*. Кар-Карыч хочет, чтобы Копатыч как можно дольше думал о горах, поэтому даст ему несколько задач. Помогите Копатычу справиться с ними! Так как ответ на задачу может быть большим, выведите его остаток от деления на $10^9 + 7$.

Перестановка — это последовательность длины n целых чисел от 1 до n , в которой все числа встречаются ровно по одному разу. Например, (1) , $(4, 3, 5, 1, 2)$, $(3, 2, 1)$ — перестановки, а $(1, 1)$, $(4, 3, 1)$, $(2, 3, 4)$ — нет.

Подотрезок — это непрерывная часть перестановки. Например, для перестановки $(2, 3, 1, 5, 4)$ $(2, 3, 1)$, (1) , $(5, 4)$ будут являться подотрезками, а $(2, 1, 5)$ и $(4, 5)$ — нет. Подотрезок является возрастающим, если числа в нём образуют возрастающую последовательность, то есть каждое число подотрезка, кроме первого, больше предыдущего. Подотрезки $(2, 5, 7)$ и (3) являются возрастающими, а подотрезки $(2, 1)$ и $(4, 5, 8, 7)$ — нет.

Формат входных данных

В первой строке входных данных содержится одно целое число t ($1 \leq t \leq 10^4$) — количество задач Кар-Карыча. Далее следуют описания задач.

Единственная строка описания каждой задачи содержит одно целое число n ($1 \leq n \leq 10^9$) — количество чисел в перестановке.

Формат выходных данных

Для каждой задачи Кар-Карыча в отдельной строке выведите остаток от деления на $10^9 + 7$ ответа на задачу.

Пример

стандартный ввод	стандартный вывод
3	0
1	5
3	499444872
200	

Замечание

Для $n = 1$ единственную перестановку (1) нельзя разбить на два непустых подотрезка.

Для $n = 3$ *новогорными* будут перестановки $(1, 2, 3)$, $(1, 3, 2)$, $(2, 1, 3)$, $(2, 3, 1)$, $(3, 1, 2)$.

Задача D. Задача из Миасса

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

По результатам замеров глубин в озере Тургояк была составлена карта. Она имеет вид прямоугольной таблицы, каждая из ячеек которой содержит либо неотрицательное целое число — глубину в каком-то месте озера, либо число -1 , означающее, что в этом месте замеры не проводились, и глубина неизвестна.

Гидрологи утверждают, что дно на измеренном участке не может иметь резких перепадов. Это означает, что глубины в ячейках, соседних по стороне, должны отличаться не более чем на v . Это касается как известных, так и неизвестных (тех, вместо которых в таблице указано число -1) глубин.

По заданной карте определите, существует ли набор неотрицательных целых глубин, точно соответствующий всем замерам, или в данных есть противоречия, и сделанные замеры были неточными.

Формат входных данных

В первой строке входных данных содержится три неотрицательных целых числа n , m и v ($1 \leq n \cdot m \leq 10^5$, $0 \leq v \leq 10^9$) — размеры карты и максимальный допустимый перепад глубин.

В следующих n строках входных данных содержатся по m целых чисел d_{ij} ($-1 \leq d_{ij} \leq 10^9$) — информация о глубинах.

Формат выходных данных

Выведите строку «Yes», если существует набор глубин, соответствующий всем замерам, и «No» иначе.

Вы можете вывести ответ в любом регистре (верхнем или нижнем). Например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительные ответы.

Примеры

стандартный ввод	стандартный вывод
3 5 2 13 -1 14 -1 10 12 -1 -1 -1 -1 14 -1 17 -1 -1	Yes
2 4 10 23 -1 -1 55 31 36 42 49	No
1 2 0 91 92	No

Замечание

В первом примере из условия подходит, например, такой набор глубин:

13	13	14	12	10
12	14	16	14	12
14	16	17	15	13

Во втором примере из условия заполнить ячейки с неизвестными глубинами невозможно. С одной стороны, неизвестная глубина рядом с 23 должна быть не больше 33. С другой стороны, неизвестная глубина рядом с 55 должна быть не меньше 45. Значит, разность между ними составляет как минимум 12 — противоречие.

В третьем примере из условия противоречие содержится в уже замеренных глубинах. Дно должно быть абсолютно ровным, но соседние глубины отличаются на 1.

Задача Е. Задача из Депутатского

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Глава посёлка Депутатский, разглядывая в очередной раз карту области, восхищался коллегами, управляющими Берлином, Парижем, Варной и Лейпцигом. И тут ему в голову пришла гениальная идея: переименовать посёлок в Манхэттен. Однако чтобы общественные обсуждения прошли гладко, он решил начать с воссоздания известной дорожной сети Манхэттена.

Итак, он поручил снести все здания в поселке и начать укладывать дорогу. Через некоторое время работа была сделана, но, начав сверять результат с генеральным планом, он пришёл в ужас!

Дорожная система сейчас представляет собой сетку размера $n \times m$: в каждом из n рядов находится по m перекрестков. Обозначим перекресток на пересечении r -й строки и c -го столбца в индексации с нуля за (r, c) . Тогда сейчас построены двусторонние дороги, соединяющие перекрёстки (i_1, j_1) и (i_2, j_2) , если $|i_1 - i_2| + |j_1 - j_2| = 1$. Также были построены двусторонние дороги, соединяющие первый и последний перекресток в каждой строке и столбце, то есть соединёнными оказались следующие пары перекрёстков: $(i, 0)$ и $(i, m - 1)$ для всех $0 \leq i < n$, а также $(0, j)$ и $(n - 1, j)$ для всех $0 \leq j < m$. Наконец, укладку дороги на некоторых перекрёстках ещё не закончили, поэтому сейчас невозможно проехать по дороге, соединяющей такой перекрёсток с любым другим.

После увиденного глава посёлка хочет задать взбучку главе стройки, поэтому сказал ехать ему навстречу. Пусть первый сейчас находится на перекрестке (i_1, j_1) , а второй — на перекрёстке (i_2, j_2) . За одну минуту они могут **одновременно** переместиться одним из следующих образом:

- Первый окажется в $((i_1 + 1) \bmod n, j_1)$, второй в $((i_2 - 1 + n) \bmod n, j_2)$.
- Первый окажется в $((i_1 - 1 + n) \bmod n, j_1)$, второй в $((i_2 + 1) \bmod n, j_2)$.
- Первый окажется в $(i_1, (j_1 + 1) \bmod m)$, второй в $(i_2, (j_2 - 1 + m) \bmod m)$.
- Первый окажется в $(i_1, (j_1 - 1 + m) \bmod m)$, второй в $(i_2, (j_2 + 1) \bmod m)$.

При этом, если хотя бы один из перекрёстков, куда они хотят переместиться, ещё не достроен, то такое перемещение невозможно.

Глава посёлка хочет понять, за какое минимальное время он сможет встретиться на одном перекрёстке с главой стройки, если будет действовать оптимально, и смогут ли они встретиться вообще.

Формат входных данных

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.

В первой строке каждого набора входных данных содержится два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — размеры дорожной системы посёлка.

В следующих n строках каждого набора входных данных содержится строка длины m , состоящая из символов «.» и «#». Если перекрёсток (i, j) достроен, то в i -й строке в j -й позиции (в индексации с нуля) находится символ «.». Если перекрёсток не достроен, то соответствующий символ равен «#».

В следующей строке каждого набора входных данных содержится два целых числа (a, b) ($0 \leq a \leq n - 1, 0 \leq b \leq m - 1$) — перекресток, на котором находится глава посёлка в начале пути.

В последней строке каждого набора входных данных содержится два целых числа (c, d) , ($0 \leq c \leq n - 1, 0 \leq d \leq m - 1$) — перекрёсток, на котором находится глава стройки в начале пути.

Гарантируется, что перекрестки (a, b) и (c, d) достроены, а также, что эти перекрестки различны.

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

Формат выходных данных

Выведите одно целое число — минимальное количество минут, по прошествии которых глава посёлка и стройки смогут встретиться. Если встреча невозможна, выведите -1 .

Пример

стандартный ввод	стандартный вывод
2 5 5 ..#.. ..#.. ##### ..#.. ..#.. 0 0 3 3 1 2 .. 0 0 0 1	2 -1

Замечание

В первом наборе входных данных первое одновременное перемещение может быть следующим: глава посёлка едет налево, проезжает по дороге, соединяющей перекрёстки $(0, 0)$ и $(0, 4)$, и оказывается на перекрёстке $(0, 4)$; глава стройки, соответственно, едет направо и оказывается на перекрёстке $(3, 4)$. В качестве второго одновременного перемещения глава посёлка едет вверх и проезжает по дороге, соединяющей перекрёстки $(0, 4)$ и $(4, 4)$, оказываясь на перекрёстке $(4, 4)$; глава стройки в это время едет вниз и тоже оказывается на перекрёстке $(4, 4)$. Таким образом они встретятся за 2 перемещения.

Во втором наборе входных данных можно показать, что глава посёлка и глава стройки никогда не встретятся.

Задача F. Задача из Копейска

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Пусть x — целое положительное число. Определим функцию $f(x)$ следующим образом:

$$f(x) = \text{lcm}(\{1, 2, 3, \dots, x-1, x\})$$

Здесь $\text{lcm}(a)$ обозначает наименьшее общее кратное множества a , то есть минимальное целое положительное число, делящееся нацело на каждый из элементов множества a .

Даны положительные целые числа l и r . Необходимо найти количество различных значений $f(i)$ среди $l \leq i \leq r$.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Далее следует их описание.

Единственная строка каждого набора входных данных содержит два целых числа l и r ($1 \leq l \leq r \leq 10^7$).

Формат выходных данных

Для каждого набора входных данных выведите количество различных значений $f(i)$ среди $l \leq i \leq r$ в отдельной строке.

Пример

стандартный ввод	стандартный вывод
2	2
11 14	5
1 5	

Замечание

В первом наборе входных данных значения $f(i)$ для $i \in [11, 14]$ равны 27720, 27720, 360360, 360360. Среди этих четырех чисел — два различных, поэтому ответ равен 2.

Во втором наборе входных данных для $i \in [1, 5]$ все значения $f(i)$ различны, поэтому ответ равен 5.

Задача G. Задача из Берлина

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В Челябинской области есть небольшое село Берлин, в котором расположено n домов, каждый из которых окружён забором. Глава Берлина решил улучшить облик села и покрасить заборы домов, на которых облупилась краска.

Дома пронумерованы целыми числами от 1 до n , и за домом с номером i идёт дом с номером $i + 1$ для каждого $1 \leq i < n$. Для каждого дома i известно целое значение a_i от нуля до единицы:

- $a_i = 0$, если забор требует покраски;
- $a_i = 1$, если забор покрашен.

Однако покраска выполняется особым образом: за один раз можно нанять бригаду из k человек, которая способна покрасить любой подотрезок заборов длины k за стоимость $cost_k$. Но бригада будет красить подотрезок заборов только в том случае, **если хотя бы один из заборов на этом подотрезке домов уже покрашен**. При этом любую бригаду можно нанять неограниченное число раз — в том числе ноль.

Более формально — если мы нанимаем бригаду размера k , то выбирается номер дома i такой, что:

- $1 \leq i \leq n - k + 1$;
- существует такой номер дома j , что $i \leq j \leq i + k - 1$ и при этом забор у дома с номером j уже покрашен.

После этого красятся все заборы для домов с номерами $i, i + 1, \dots, i + k - 1$.

Глава села хочет потратить как можно меньше денег, поэтому просит вас по заданным входным данным найти минимальную суммарную стоимость, чтобы покрасить **все заборы**.

Формат входных данных

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.

В первой строке каждого набора входных данных содержится одно целое число n ($1 \leq n \leq 10^5$) — количество домов в Берлине.

Во второй строке каждого набора входных данных содержится n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 1$), где a_i обозначает состояние забора у дома с номером i .

В третьей строке каждого набора входных данных содержится n целых чисел $cost_1, cost_2, \dots, cost_n$ ($1 \leq cost_i \leq 100$), где $cost_i$ — стоимость покраски подотрезка домов длины i .

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$. Также гарантируется, что в каждом наборе входных данных есть хотя бы один покрашенный забор.

Формат выходных данных

Для каждого набора выходных данных в единственной строке выведите минимальную стоимость покраски всех заборов.

Пример

стандартный ввод	стандартный вывод
6	0
1	1
1	3
1	52
3	2
1 0 1	5
3 2 1	
7	
0 0 0 1 0 0 0	
2 28 1 76 69 13 77	
2	
1 0	
99 52	
3	
0 0 1	
2 1 3	
8	
0 0 0 1 0 0 0 1	
2 2 2 3 3 3 100 100	

Замечание

В первом наборе входных данных все заборы уже покрашены, поэтому ответ будет равен нулю.

Во втором наборе входных данных можно нанять бригаду из трёх человек, выбрав $i = 1$.

В третьем наборе входных данных можно трижды нанять бригаду из трёх человек: сначала выбрав $i = 3$, затем $i = 1$, и в конце выбрав $i = 5$.

В пятом наборе входных данных можно дважды нанять бригаду из двух человек: сначала выбрав $i = 2$, а затем выбрав $i = 1$.

Задача Н. Задача из Златоуста

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дана матрица целых чисел, состоящая из n строк и m столбцов. Строки матрицы пронумерованы целыми числами от 1 до n , столбцы целыми числами от 1 до m . За одну операцию можно выбрать любую строку i и любой столбец j и удалить из матрицы все числа, находящиеся в i -й строке или j -м столбце. Все остальные элементы при этом сдвигаются, заполняя пропуски.

Более формально, если до операции число в x -й строке и y -м столбце было $a_{x,y}$, то после операции оно сдвинется в предыдущую строку, если $x > i$, сдвинется в предыдущий столбец, если $y > j$, а если $x = i$ или $y = j$, то будет удалено.

Операцию нельзя применять, если в матрице осталась всего одна строка или один столбец. Определите, можно ли этими операциями добиться того, чтобы в каждой строке и в каждом столбце матрицы числа шли по возрастанию.

Формат входных данных

В первой строке входных данных содержится два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^6$, $n \cdot m \leq 2 \cdot 10^6$) — размеры матрицы.

В следующих n строках входных данных содержится по m целых чисел: j -е число в i -й строке — это $a_{i,j}$ ($0 \leq a_{i,j} \leq 10^9$), число, изначально записанное в i -й строке и в j -м столбце матрицы.

Формат выходных данных

Выведите «Yes», если можно добиться того, чтобы в каждой строке и в каждом столбце числа шли по возрастанию, и «No» иначе.

Вы можете вывести ответ в любом регистре (верхнем или нижнем). Например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительные ответы.

Примеры

стандартный ввод	стандартный вывод
2 4 1 15 20 10 33 27 6 18	Yes
1 1 1000000000	Yes
2 5 0 0 0 1 2 6 7 8 2 3	No

Замечание

В первом примере из условия можно применить операцию для второй строки и четвёртого столбца. В результате в матрице останется одна строка с числами 1, 15, 20.

Во втором примере из условия в матрице всего одно число, поэтому можно сказать, что в каждом столбце и в каждой строке числа уже возрастают.

В третьем примере из условия добиться желаемого невозможно вне зависимости от того, как применять операции. Обратите внимание, что числа в итоговой матрице обязаны возрастать строго, равные числа в одной строке или в одном столбце не допускаются.

Задача I. Задача из Малиновки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Смешарики поехали на летний отдых в деревню Малиновку. Чтобы подсластить досуг, Совунья подарила Нюше набор конфет, состоящий из a конфет «Алёнка», b конфет «Белочка» и c конфет «Цитрон». Гарантируется, что суммарное количество конфет чётное, т.е. $a + b + c$ делится на 2, и $\frac{a+b+c}{2} = n$.

Бараш придумал игру для Нюши: каждый день в течение n дней он будет приходить к Нюше и говорить ей выбрать любые две конфеты из ещё несъеденных и съесть их. Нюша узнала желания Бараша и выразила их строкой s длины n , состоящей из символов «0» и «1». Если $s_i = \text{«0»}$, то в i -й день Бараш будет дарить Нюше розу, только если две выбранные конфеты будут одинаковые, а если $s_i = \text{«1»}$, то в i -й день Бараш будет дарить Нюше розу, только если две выбранные конфеты будут разные.

Нюша хочет получить наибольшее возможное количество роз от Бараша, поэтому она просит вас помочь ей выбрать конфеты в каждый из n дней так, чтобы максимизировать количество подаренных ей роз.

Формат входных данных

В первой строке входных данных содержится три целых числа a , b и c ($0 \leq a, b, c \leq 2 \cdot 10^5$, $a + b + c = 2n$, $1 \leq n \leq 10^5$) — количество конфет «Алёнка», «Белочка» и «Цитрон» соответственно.

Во второй строке входных данных содержится строка s длины n , состоящая из символов «0» и «1». $s_i = \text{«0»}$, если в i -й день Бараш подарит Нюше розу за две одинаковые конфеты, и $s_i = \text{«1»}$, если за две разные конфеты.

Обратите внимание, что число n явно не даётся во входных данных.

Формат выходных данных

В первой строке выходных данных выведите единственное целое число — максимальное количество роз, которое Нюша сможет получить.

В каждой из следующих n строк выведите по два символа английского алфавита «a», «b» или «c» через пробел — в i -й строке выведите символы, обозначающие две конфеты, которые стоит выбрать Нюше в i -й день. Символ «a» означает выбор конфеты «Алёнка», символ «b» означает выбор конфеты «Белочка», символ «c» означает выбор конфеты «Цитрон».

Заметьте, что в выводе символ «a» должен встретиться ровно a раз, символ «b» ровно b раз, и символ «c» ровно c раз.

Два символа, обозначающие съеденные за один день конфеты, можно выводить в любом порядке. Если существует несколько способов съесть конфеты так, чтобы получить максимальное количество роз, выведите любой.

Примеры

стандартный ввод	стандартный вывод
3 3 2 0101	4 a a b c b b a c
8 0 0 1111	0 a a a a a a a a

Задача J. Задача из Челябинска

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Челябинске открыли новую обсерваторию, и Лосяш переехал туда, чтобы заниматься астрономией.

Лосяш наблюдает за n космическими объектами, расположенными подряд друг за другом в одном ряду. i -й по счёту объект по наблюдениям Лосяша имеет целочисленную массу a_i . Космические объекты входят в m галактик, где каждая галактика является непрерывным отрезком космических объектов. Более формально, j -ю галактику можно представить с помощью двух чисел l_j и r_j ($1 \leq l_j \leq r_j \leq n$), обозначающих, что все объекты с номерами i такими, что $l_j \leq i \leq r_j$, входят в j -ю галактику. Заметьте, что галактики могут пересекаться между собой.

Массой галактики является сумма всех объектов, входящих в неё, т.е. масса галактики j будет равна $\sum_{i=l_j}^{r_j} a_i$. *Масса Вселенной* равна максимальной массе из m галактик, за которыми наблюдает Лосяш.

Лосяш знает, что один из наблюдаемых космических объектов является чёрной дырой, масса которой может быть абсолютно любой. Другими словами, существует **ровно один** космический объект $1 \leq i \leq n$, что его масса может быть абсолютно любой (в том числе и бесконечно маленькой), а также может отличаться от значения a_i .

Коллега Лосяша подсказала ему, что масса и расположение чёрной дыры являются такими, чтобы *масса Вселенной* была минимально возможной. Зная, что один из n наблюдаемых объектов на самом деле является чёрной дырой и может иметь любую массу, помогите Лосяшу узнать *массу Вселенной*.

Формат входных данных

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

В первой строке каждого наблюдения содержится два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — количество космических объектов и галактик соответственно.

Во второй строке каждого наблюдения содержится n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — предполагаемые массы космических объектов.

В следующих m строках каждого наблюдения содержится по два целых числа, в j -й строке находятся числа l_j и r_j ($1 \leq l_j \leq r_j \leq n$) — границы j -й галактики.

Гарантируется, что сумма n и сумма m по всем наблюдениям не превышает $2 \cdot 10^5$.

Формат выходных данных

Для каждого наблюдения Лосяша в отдельной строке выведите минимально возможную массу Вселенной, если она конечна, либо строку `-inf`, если масса Вселенной бесконечно маленькая.

Пример

стандартный ввод	стандартный вывод
2	-14
5 3	-inf
-7 -5 -9 -5 5	
1 3	
2 3	
4 5	
4 1	
1 -1 4 3	
1 3	

Задача К. Задача из Озёрска

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Иван почти доделал ремонт в ванной комнате. Хотя, если говорить честно, пока что он успел только положить плитку на пол. Узор каждой плитки представляет собой клетчатое поле размера $h \times w$. Каждая клетка окрашена в чёрный или белый цвет. Пол в ванной Ивана устроен таким образом, что в него уместилось ровно n плиток, образовав клетчатое поле размера $h \times nw$.

Рассмотрим это клетчатое поле сверху: пронумеруем строки сверху вниз числами от 1 до h , а столбцы слева направо числами от 1 до nw . Координатами клетки будем считать два числа (r, c) , где r — номер строки, а c — номер столбца.

Так как в рамках ремонта Иван сделал только плитку, ему стало грустно, и он решил развлечься. Он q раз выбрал две белые клетки, расположенные в клетках с координатами (r_1, c_1) и (r_2, c_2) . Для каждой выбранной пары клеток он хочет узнать, существует ли путь из первой во вторую, проходящий строго по белым клеткам. Соседние клетки в пути должны иметь общую сторону. При этом есть очень важное условие, что **зазор между двумя плитками можно пересекать строго в направлении слева направо**. В пределах одной плитки можно двигаться по белым клеткам в любом направлении.

Помогите Ивану ответить на все запросы. Гарантируется, что все клетки, данные в запросах, окрашены в белый цвет.

Формат входных данных

В первой строке входных данных содержится три целых числа h, w, n ($1 \leq h, w \leq 250$, $1 \leq n \leq 10^9$) — высоту, ширину узора плитки и количество плиток.

В следующих h строках входных данных описывается узор плитки. Каждая из этих h строк содержит w символов. В r -й строке c -й символ равен «0», если клетка с координатами (r, c) является чёрной, и «1» — если белой.

В следующей строке входных данных содержится одно целое число q ($1 \leq q \leq 10^6$) — количество запросов.

В следующих q строках входных данных описываются запросы — четыре целых числа r_1, c_1, r_2 и c_2 ($1 \leq r_1, r_2 \leq h$, $1 \leq c_1, c_2 \leq nw$).

Гарантируется, что все клетки (r_1, c_1) и все клетки (r_2, c_2) окрашены в белый цвет.

Формат выходных данных

Для каждого запроса выведите ответ на задачу — выведите «Yes», если из ячейки (r_1, c_1) можно добраться до ячейки (r_2, c_2) , перемещаясь только по белым ячейкам, и «No» иначе.

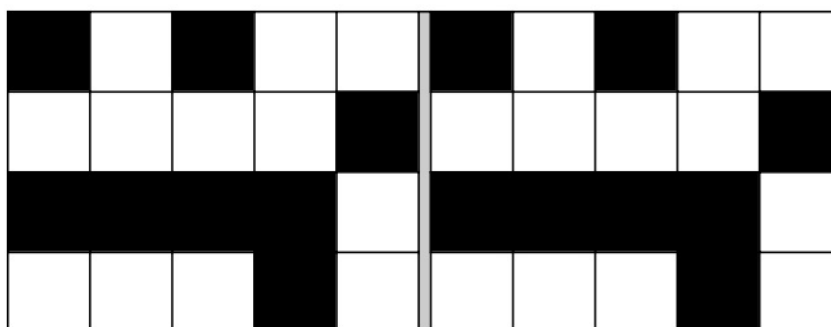
Вы можете вывести ответ в любом регистре (верхнем или нижнем). Например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительные ответы.

Примеры

стандартный ввод	стандартный вывод
4 5 2 01011 11110 00001 11101 6 2 1 1 5 1 4 1 9 1 4 2 3 3 5 4 7 2 2 3 5 4 7 3 5	Yes No Yes Yes No No
1 2 4 11 3 1 1 1 8 1 5 1 4 1 6 1 7	Yes No Yes

Замечание

Узор плитки из первого примера приведён на рисунке ниже.



Задача L. Задача из Катав-Ивановска

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Мальчик Петя очень любит игровые автоматы. Сегодня он решил поиграть на одном из них.

Автомат имеет три барабана по n ячеек, в каждой из которых записано по одному целому числу. На первом барабане записаны числа $a_1, a_2, \dots, a_n$, на втором — $b_1, b_2, \dots, b_n$; на третьем — $c_1, c_2, \dots, c_n$. В любой момент времени на каждом из барабанов видно только одно число из записанных. В начальный момент времени на каждом из барабанов видны первые элементы, то есть значения a_1, b_1 и c_1 .

Одна игра с таким автоматом стоит $cost$ монет. После оплаты барабаны поворачиваются циклически (после ячейки с номером n снова идет первая ячейка): первый на x ячеек вправо, второй на y , третий на z .

После поворота барабанов выполняется проверка выигрыша. Если на **всех** барабанах видно одно и то же число, то игрок получает выигрыш, который равен значению этого числа, иначе не получает ничего.

Петя может прекратить игру в любой момент, в том числе не начав играть вовсе, но он будет играть не больше q раз. Петя знает все числа на барабанах, величины поворотов, но не знает, как посчитать максимальную прибыль. Прибыль — это разница между суммарным выигрышем и стоимостью совершённых игр.

Помогите Пете определить максимальную прибыль, которую он может получить.

Формат входных данных

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.

В первой строке каждого набора входных данных содержится три целых числа n , $cost$ и q ($1 \leq n \leq 10^5$, $0 \leq cost \leq 10^9$, $1 \leq q \leq 10^9$) — размер барабанов, стоимость одной игры и максимальное число игр соответственно.

Во второй строке каждого набора входных данных содержится три целых числа x , y и z ($1 \leq x, y, z \leq 10^9$) — величины поворотов барабанов.

В третьей строке каждого набора входных данных содержится n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — числа в ячейках первого барабана. Перед началом первой игры на барабане видно число a_1 , остальные числа перечислены в порядке поворота барабана.

В четвёртой строке каждого набора входных данных содержится n целых чисел $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq 10^9$) — числа в ячейках второго барабана. Перед началом первой игры на барабане видно число b_1 , остальные числа перечислены в порядке поворота барабана.

В пятой строке входных данных содержится n целых чисел $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq 10^9$) — числа в ячейках третьего барабана. Перед началом первой игры на барабане видно число c_1 , остальные числа перечислены в порядке поворота барабана.

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

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — максимальную возможную прибыль Пети.

Пример

стандартный ввод	стандартный вывод
4	12
10 1 3	2
5 3 4	0
14 11 11 11 10 13 10 14 12 14	3
10 12 10 13 10 12 14 11 13 12	
12 11 12 11 10 14 11 11 14 13	
10 0 9	
10 7 4	
2 3 3 3 4 3 1 1 5 2	
2 5 1 2 2 5 1 5 3 1	
1 3 1 1 5 2 2 1 1 5	
3 1 100	
1 1 1	
3 3 3	
7 7 7	
8 8 8	
1 1 3	
1 1 1	
2	
2	
2	

Замечание

В первом наборе входных данных максимальная прибыль достигается после второй игры. После первой игры будут видны числа a_6 , b_4 и c_5 . Но они не равны между собой, поэтому выигрыша не будет. После второй игры будут видны числа a_1 , b_7 и c_9 . Все эти числа равны между собой, поэтому выигрыш будет равен 14. Суммарные затраты на две игры будут равны $1 \cdot 2 = 2$.

Во втором наборе входных данных максимальная прибыль достигается после последней, 9-й игры. Закончив игру раньше, Петя остался бы вообще без выигрыша.

В третьем наборе входных данных Пете выгодно даже не начинать игру. Поскольку на барабанах вообще нет совпадающих чисел, выигрыша он никогда не получит.

Задача М. Задача из Черёмушек

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Хошино Аква — гениальный ребёнок-актёр из знаменитого посёлка Черёмушки. Недавно он вместе со своей сестрой Руби переехал в Токио, чтобы помочь ей стать великим айдолом.

Руби уже состоит в айдол группе «В-Komachi», вместе с Каной и Мем-чо. Группа готовится к серии из t выступлений. Каждое выступление будет проведено на сцене, представляющей из себя нестрого выпуклый многоугольник с n вершинами, то есть такой, в котором некоторые вершины могут лежать на сторонах, а не только в углах.

Аква быстро заметил, что вершины сцены — это крепления для кабелей студийной аппаратуры. Он решил провести два кабеля через всю сцену, соединяя вершины, чтобы разбить её на три отдельные секции — по одной для каждой участницы. Однако просто разбить недостаточно для успеха, Аква хочет, чтобы площади трёх секций относились как $a : b : c$.

Поскольку Аква — актёр, а не программист, вам, как менеджеру «В-Komachi», предстоит посчитать количество способов провести два кабеля, чтобы сцена разделилась на три секции с заданным отношением площадей. Только так группа сможет достичь звёздного успеха на каждом выступлении!

Формат входных данных

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

Для каждого выступления:

- В первой строке содержится одно целое число n ($5 \leq n \leq 2 \cdot 10^5$) — количество вершин сцены.
- В следующих n строках содержится по два целых числа x_i и y_i ($0 \leq |x_i|, |y_i| \leq 10^6$) — координаты вершин сцены в порядке обхода.

Гарантируется, что любая пара различных вершин не совпадает.

- В последней строке содержится три целых числа a , b и c ($1 \leq a, b, c \leq 10^9$) — желаемое отношение площадей трёх секций.

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

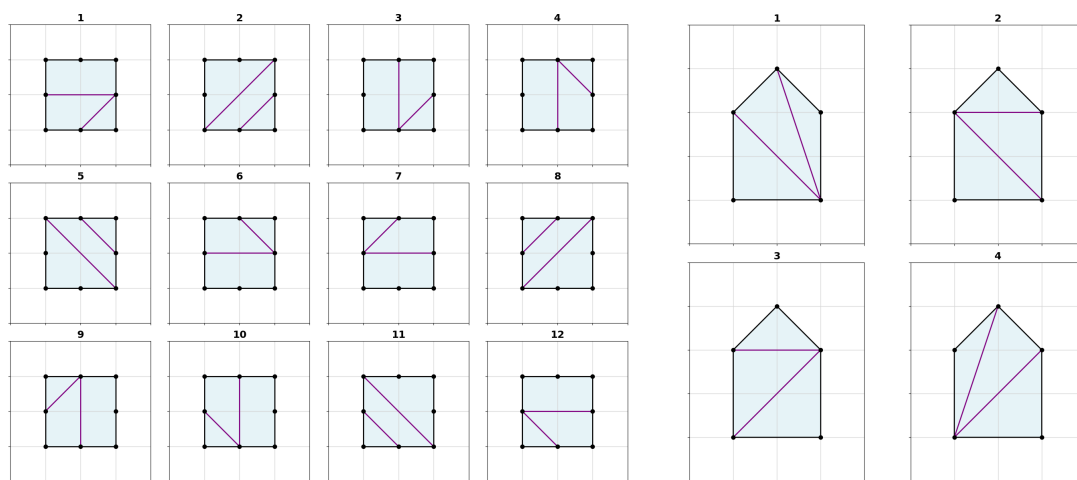
Формат выходных данных

Для каждого выступления выведите одно целое число — количество способов провести два кабеля, чтобы сцена разделилась на три секции с отношением площадей $a : b : c$.

Примеры

стандартный ввод	стандартный вывод
2 8 0 0 1 0 2 0 2 1 2 2 1 2 0 2 0 1 1 3 4 5 0 0 2 0 2 2 1 3 0 2 2 4 4	12 4
1 8 0 0 0 5 9 5 10 5 12 5 12 0 10 0 9 0 45 5 10	3

Замечание



Все варианты расположения кабелей для двух выступлений в первом примере из условия.

Задача N. Задача из Коркино

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дана строка $s_1s_2\dots s_n$, состоящая из букв **a** и **b**. Ваша задача ответить на следующие q запросов к этой строке:

- В каждом запросе вам даётся два целых числа $1 \leq l \leq r \leq n$. Ваша задача проверить, отсортирована ли подстрока $s_ls_{l+1}\dots s_r$ по неубыванию.

Формат входных данных

В первой строке входных данных содержится два целых числа n и q ($1 \leq n, q \leq 2 \cdot 10^5$) — длина строки s и количество запросов.

Во второй строке входных данных содержится строка s длины n , состоящая из букв **a** и **b**.

В следующих q строках входных данных описываются запросы. Каждая строка содержит два целых числа l и r ($1 \leq l, r \leq n$).

Формат выходных данных

Для каждого запроса выведите «Yes», если подстрока отсортирована по неубыванию, иначе «No».

Вы можете вывести ответ в любом регистре (верхнем или нижнем). Например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительные ответы.

Пример

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

Задача О. Задача из Кузнецкого

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У Алисы есть упорядоченный по неубыванию массив целых чисел $a_1, a_2, \dots, a_n$. Другими словами, изначально для всех $1 \leq i < n$ верно, что $a_i \leq a_{i+1}$.

Алиса **ровно один раз** выбрала некоторый подотрезок этого массива (возможно, пустой) и перемешала элементы в нём. Например, если $a = [1, 2, 3, 4, 5]$, то Алиса могла выбрать подотрезок элементов $[a_2, a_3, a_4]$ и как угодно перемешать эти элементы между собой местами. Тогда у неё может получиться любой из шести следующих массивов: $[1, 2, 3, 4, 5]$, $[1, 2, 4, 3, 5]$, $[1, 3, 2, 4, 5]$, $[1, 3, 4, 2, 5]$, $[1, 4, 2, 3, 5]$, $[1, 4, 3, 2, 5]$. Если Алиса выбирает пустой подотрезок, то массив не изменяется.

Боб не видел изначального массива и хочет понять, каким мог быть наибольший и наименьший размер выбранного Алисой подотрезка. Помогите ему решить эту задачу.

Формат входных данных

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.

В первой строке каждого набора входных данных содержится одно целое число n ($1 \leq n \leq 10^5$) — количество элементов в массиве.

Во второй строке каждого набора содержится n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — элементы массива Алисы после того, как она перемешала какой-то его подотрезок.

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

Формат выходных данных

Для каждого набора входных данных выведите через пробел два целых числа — наименьший и наибольший размер подотрезка, выбранного Алисой.

Пример

стандартный ввод	стандартный вывод
4	5 5
5	0 3
5 3 4 2 1	0 1
3	4 6
1 2 3	
1	
1	
6	
1 3 2 2 1 3	

Замечание

В первом наборе входных данных минимальное число стоит последним, а максимальное — первым, поэтому размеры наименьшего и наибольшего подотрезков совпадают и равны 5.

Во втором наборе входных данных все числа стоят на своём месте, поэтому наименьший возможный подотрезок имеет размер 0, наибольший же размер 3.