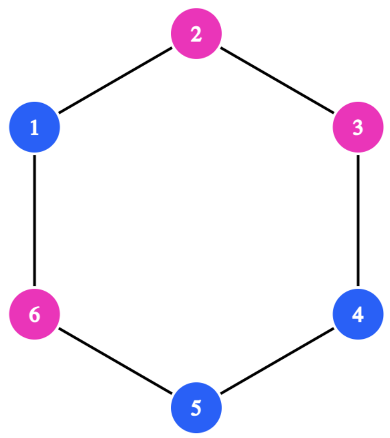
Задача A. Хоровод

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

Есть группа из x мальчиков и y девочек. Рассмотрим хоровод, составленный из них, в котором дети пронумерованы от 1 до $x + y$ по часовой стрелке.



На рисунке выше изображён хоровод из трёх мальчиков (номера 1, 4 и 5) и трёх девочек (номера 2, 3 и 6). Этот хоровод можно описать строкой `BGGBBG`, где i-й символ обозначает пол человека на i-й позиции в хороводе. Символ `B` обозначает мальчика, символ `G` обозначает девочку.

Определите, можно ли составить хоровод из всех мальчиков и девочек в группе, в котором есть ровно m пар соседей разного пола. Например, на рисунке выше есть 4 такие пары: $(1, 2)$, $(3, 4)$, $(5, 6)$ и $(6, 1)$.

Формат входных данных

Первая строка содержит одно целое число x ($2 \leqslant x \leqslant 100$) — количество мальчиков в хороводе.

Вторая строка содержит одно целое число y ($2 \leqslant y \leqslant 100$) — количество девочек в хороводе.

Третья строка содержит одно целое число m ($2 \leqslant m \leqslant 200$) — необходимое количество пар соседей разного пола.

Формат выходных данных

В первой строке выведите «`YES`» (без кавычек), если мальчиков и девочек можно расставить в хороводе требуемым образом, и «`NO`» (без кавычек) иначе.

Если требуемая расстановка существует, во второй строке выведите строку длины $x + y$, содержащую x символов `B` (мальчик) и y символов `G` (девочка). Эта строка должна описывать любой хоровод, в котором есть ровно m пар соседей разного пола. Обратите внимание, что первый и последний символ строки описывают соседних детей в хороводе.

Система оценки

В данной задаче 20 тестов, помимо тестов из условия, каждый из них оценивается в 5 баллов. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $x = y$, наберут не менее 30 баллов.

Примеры

стандартный ввод	стандартный вывод
3 3 4	YES BGGBBG
2 2 3	NO
3 2 4	YES BGBGB
2 2 100	NO

Замечание

Хоровод из первого примера изображён в условии задачи.

Во втором примере не существует хоровода из 2 мальчиков и 2 девочек, в котором есть ровно 3 пары соседей разного пола.

В третьем примере есть ровно 4 пары соседей разного пола: $(1, 2)$, $(2, 3)$, $(3, 4)$, $(4, 5)$.

Задача B. Суммы из отрезка

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

Петя работает продавцом в лавке чисел. В наличии имеется бесконечное количество каждого из чисел $l, l{+}1, \ldots, r$ (при этом $l < r$). Когда в магазине нет покупателей, Петя от скуки берет некоторое количество чисел и считает их сумму. Обратите внимание, что Петя может брать одинаковые числа сколько угодно раз. При этом могут существовать некоторые целые положительные числа, которые он никогда не сможет получить в качестве суммы. Помогите Пете найти **максимальное** такое число.

Другими словами, найдите максимальное целое положительное число, которое нельзя представить в виде суммы произвольного количества целых чисел из отрезка $[l, r]$. Если с помощью этих чисел можно получить любое целое положительное число, выведите -1.

Формат входных данных

Первая строка содержит одно целое число l ($1 \leqslant l \leqslant 10^9$) — левая граница отрезка.

Вторая строка содержит одно целое число r ($1 \leqslant r \leqslant 10^9$) — правая граница отрезка.

Гарантируется, что $l < r$.

Формат выходных данных

Выведите одно целое число — максимальное положительное число, которое нельзя получить в виде суммы чисел, принадлежащих отрезку. Если такого числа нет, выведите -1.

Обратите внимание, что ответ может быть больше, чем возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип int64 в языке Pascal, тип long long в C и C++, тип long в Java и C#). Язык Python будет корректно работать.

Система оценки

В данной задаче 20 тестов, помимо тестов из условия, каждый из них оценивается в 5 баллов. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $r \leqslant 100$, наберут не менее 20 баллов.

Решения, корректно работающие при $r \leqslant 10^5$, наберут не менее 50 баллов.

Примеры

стандартный ввод	стандартный вывод
4 5	11
6 8	17
1 10	-1

Замечание

В первом примере максимальное число, которое нельзя получить — 11. Можно показать, что все числа больше 11 получить можно. Например, 13 можно получить как $4 + 4 + 5$.

Во втором примере число 17 получить нельзя, а все большие — можно.

В третьем примере в виде суммы чисел из отрезка можно представить любое целое положительное число.

Задача C. Крайний Банк

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

Крайний банк занимается важной деятельностью — контролем биржей бипок и обнаружением потенциального мошенничества. Вам, как стажеру Крайнего Банка, поручили разработать систему, которая позволяла бы определять, является ли данный трейдер потенциальным мошенником или нет. Вы знаете, что биржа бипок работала n моментов времени, в течение которых трейдер торговал на бирже.

Вам известно, что в i-й момент времени цена одной бипки составляла a_i и трейдер продал или купил $|b_i|$ бипок по данной цене. Если $b_i < 0$, то он продал $-b_i$ бипок, а если $b_i > 0$, то купил b_i бипок. На момент начала торгов у трейдера было 0 бипок, а также известно, что к концу торгов он распродал все бипки (другими словами, сумма b_i равна 0). Также трейдер не мог уходить в минус, то есть в любой момент времени у него было неотрицательное количество бипок. Если $b_i = 0$, то в i-й день трейдер не продавал и не покупал бипки.

Крайний банк считает трейдера потенциальным мошенником, если каждой операции покупки **одной конкретной бипки** можно сопоставить в пару операцию продажи одной бипки так, что покупка произошла раньше продажи и стоимость покупки **строго меньше** стоимости продажи. Обратите внимание, что разным операциям покупки одной бипки должны соответствовать **разные** операции продажи. При этом, если две бипки были куплены в один и тот же момент времени, операции продажи для них могли произойти в разное время.

Определите, является ли данный трейдер потенциальным мошенником.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t ($1 \leqslant t \leqslant 10\,000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($2 \leqslant n \leqslant 200\,000$) — количество моментов времени, в течение которых трейдер торговал на бирже.

Вторая строка каждого набора входных данных содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — стоимость одной бипки в i-й момент времени.

Третья строка каждого набора входных данных содержит n целых чисел $b_1, b_2, \ldots, b_n$ ($-10^9 \leqslant b_i \leqslant 10^9$) — числа, характеризующие операции покупки/продажи, совершенные трейдером, в соответствии с форматом, описанным в условии.

Гарантируется, что в любой момент времени у трейдера было неотрицательное количество бипок, а также $b_1 + b_2 + \ldots + b_n = 0$.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $200\,000$.

Формат выходных данных

Для каждого набора входных данных выведите «`Yes`» (без кавычек), если трейдер является потенциальным мошенником, иначе выведите «`No`» (без кавычек).

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

Система оценки

В данной задаче 25 тестов, помимо тестов из условия, каждый из них оценивается в 4 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Обозначим за N сумму n по всем наборам входных данных в данном тесте.

Решения, корректно работающие при $n \leqslant 7$, $t \leqslant 10$, наберут не менее 24 баллов.

Решения, корректно работающие при $N \leqslant 3000$, наберут не менее 60 баллов.

Решения, корректно работающие при $a_i \leqslant 3$, наберут не менее 28 баллов.

Пример

стандартный ввод	стандартный вывод
4 3 2 3 2 1 1 -2 3 2 3 4 1 1 -2 5 2 3 3 3 4 1 1 -1 1 -2 2 1000000000 1000000000 1000000000 -1000000000	No Yes Yes No

Замечание

В первом наборе входных данных трейдер продает бипки, купленные по цене 2 и 3 по цене 2. Тем самым он не выходит в плюс по каждой бипке и не является потенциальным мошенником.

Во втором наборе входных данных трейдер продает бипки, купленные по цене 2 и 3 по цене 4. Тем самым он выходит в плюс по каждой бипке и является потенциальным мошенником.

В третьем наборе входных данных трейдер может продать в третий момент времени бипку, купленную по цене 2 за 3 и выйти по ней в плюс. После чего в пятый момент времени продать все остальные бипки и выйти по ним в плюс. Таким образом, он является потенциальным мошенником.

Задача D. Максимальная попарная разница

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Вам даны два массива чисел длины n: $a_1, a_2, \ldots, a_n$ и $b_1, b_2, \ldots, b_n$. При этом известно, что все числа $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ попарно различны.

Назовём *счётом* перестановки $p_1, p_2, \ldots, p_n$ длины n значение следующего выражения: $|a_1 - b_{p_1}| + |a_2 - b_{p_2}| + \ldots + |a_n - b_{p_n}|$.

Вам нужно найти максимально возможный счёт по всем перестановкам длины n, а также количество перестановок длины n, имеющих этот максимальный счёт. Так как количество перестановок может быть достаточно большим, требуется найти только его остаток при делении на $10^9 + 7$. Обратите внимание, что от самого максимально возможного счёта **не нужно** брать остаток.

Напомним, что перестановкой длины n является массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — перестановка, но $[1, 2, 2]$ не перестановка (2 встречается в массиве дважды) и $[1, 3, 4]$ тоже не перестановка ($n = 3$, но в массиве встречается 4).

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 50\,000$) — длина массивов a и b.

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

Третья строка содержит n целых чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant 10^9$) — элементы массива b.

Гарантируется, что все числа $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ попарно различны. То есть, для любых $1 \leqslant i, j \leqslant n$ верно, что $a_i \neq b_j$, а также $a_i \neq a_j$ и $b_i \neq b_j$ при $i \neq j$.

Формат выходных данных

Выведите два целых числа — максимально возможный счёт и количество перестановок с этим счётом. Количество перестановок выведите по модулю $10^9 + 7$. От самого максимально возможного счёта **не нужно** брать остаток.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $n \leqslant 10$, наберут не менее 16 баллов.

Решения, корректно работающие при $n \leqslant 18$, наберут не менее 30 баллов.

Решения, корректно работающие при $n \leqslant 1000$, наберут не менее 50 баллов.

Дополнительно, решения, корректно работающие при $b_i = a_i + 1$ для всех $1 \leqslant i \leqslant n$, наберут не менее 14 баллов.

Дополнительно, решения, корректно работающие при $b_{i+1} = b_i + 1$ для всех $1 \leqslant i < n$, наберут не менее 14 баллов.

Примеры

стандартный ввод	стандартный вывод
2 1 3 2 4	4 1
3 10 15 6 7 12 14	18 2
2 1 7 1000000000 998244353	1998244345 2
5 82 494 27 39390 999999999 83 495 28 39391 1000000000	2000078561 12
8 10 20 30 40 50 60 70 80 32 33 34 35 36 37 38 39	184 720

Замечание

Рассмотрим первый пример. Всего есть 2 перестановки длины 2:

1. $p = [1, 2]$. Счёт перестановки равен $|a_1 - b_1| + |a_2 - b_2| = |1 - 2| + |3 - 4| = 2$.
2. $p = [2, 1]$. Счёт перестановки равен $|a_1 - b_2| + |a_2 - b_1| = |1 - 4| + |3 - 2| = 4$.

   Максимальный счёт равен 4, и он достигается только на одной перестановке.

Рассмотрим второй пример. Всего есть 6 перестановок длины 3:

1. $p = [1, 2, 3]$. Счёт перестановки равен $|10 - 7| + |15 - 12| + |6 - 14| = 14$.
2. $p = [1, 3, 2]$. Счёт перестановки равен $|10 - 7| + |15 - 14| + |6 - 12| = 10$.
3. $p = [2, 1, 3]$. Счёт перестановки равен $|10 - 12| + |15 - 7| + |6 - 14| = 18$.
4. $p = [2, 3, 1]$. Счёт перестановки равен $|10 - 12| + |15 - 14| + |6 - 7| = 4$.
5. $p = [3, 1, 2]$. Счёт перестановки равен $|10 - 14| + |15 - 7| + |6 - 12| = 18$.
6. $p = [3, 2, 1]$. Счёт перестановки равен $|10 - 14| + |15 - 12| + |6 - 7| = 8$.

Максимальный счёт равен 18, и есть 2 перестановки, на которых достигается этот счёт.

Задача E. Рекордное удаление

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

Назовём *рекордом* элемент массива, который является максимумом на префиксе, заканчивающемся в этом элементе. Более формально, элемент a_i является рекордом в массиве a, если $a_j < a_i$ для каждого $j < i$.

Рассмотрим следующий процесс на некотором массиве a:

- На первом этапе выберем в массиве a все элементы, которые являются рекордами, и удалим их из a. При этом порядок оставшихся элементов не меняется.
- На втором этапе выберем в получившемся массиве a все элементы, которые теперь являются рекордами, и удалим их из a.
- ...
- На m-м этапе выберем в a все элементы, которые стали рекордами после предыдущего шага, и удалим их из a. Будем повторять так до тех пор, пока массив a не станет пустым.

Вам дана перестановка p длины n и q запросов, описываемые двумя числами l и k ($1 \leqslant l \leqslant n$, $1 \leqslant k \leqslant 20$). Для такого запроса мы рассмотрим массив $a = [p_l, p_{l+1}, p_{l+2}, \ldots, p_n]$ и выполним ровно k этапов удаления (если массив a станет пустым раньше, чем за k этапов, то он останется пустым). Вам требуется посчитать количество элементов, удалённых из массива a за все эти этапы. Обратите внимание, что сам массив p никак не меняется (то есть удаления в одном запросе никак не влияют на другие запросы).

Напомним, что перестановкой длины n является массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — перестановка, но $[1, 2, 2]$ не перестановка (2 встречается в массиве дважды) и $[1, 3, 4]$ тоже не перестановка ($n = 3$, но в массиве встречается 4).

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leqslant n, q \leqslant 10^5$) — длина перестановки и количество запросов.

Вторая строка содержит n различных целых чисел $p_1, p_2, \ldots, p_n$ ($1 \leqslant p_i \leqslant n$) — элементы перестановки p.

Каждая из следующих q строк содержит два целых числа l и k ($1 \leqslant l \leqslant n$, $1 \leqslant k \leqslant 20$) — описание очередного запроса.

Формат выходных данных

Для каждого запроса выведите одно целое число — количество элементов, удалённых из соответствующего массива за все этапы.

Система оценки

Тесты к этой задаче состоят из примеров и 5 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Группы в данной задаче «склеиваются». Это означает, что за подгруппу начисляются баллы, если хотя бы одна из ваших посылок успешно проходит все её тесты. Обратите внимание, что если у подгруппы имеются необходимые подгруппы, то для проверки на данной подгруппе текущая посылка должна пройти тесты во всех этих необходимых подгруппах (при этом результаты предыдущих посылок не учитываются).

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		n	q	k		
0	0	–	–	–	–	Тесты из условия
1	17	$n \leqslant 100$	$q \leqslant 100$	$k \leqslant 10$	–	–
2	12	–	–	–	–	$p_i = n - i + 1$
3	22	–	–	$k = 1$	–	–
4	21	–	–	$k \leqslant 2$	3	–
5	28	–	–	–	0 – 4	–

Примеры

стандартный ввод	стандартный вывод
3 2 2 3 1 2 2 1 1	2 2
4 3 4 3 2 1 4 1 3 3 1 3	1 2 3
7 6 3 5 1 4 6 2 7 1 1 1 2 1 3 3 3 3 1 6 20	4 6 7 5 4 2

Замечание

Рассмотрим первый пример:

- В первом запросе $l = 2$ и $k = 2$, $a = [3, 1]$. На первом этапе удалится элемент 3, после этого $a = [1]$. На втором этапе удалится элемент 1, после этого $a = []$. Таким образом, за 2 этапа удалилось 2 элемента.
- Во втором запросе $l = 1$ и $k = 1$, $a = [2, 3, 1]$. На первом этапе удалятся элементы 2 и 3, после этого $a = [1]$. Таким образом, за 1 этап удалилось 2 элемента.

Рассмотрим второй пример:

- В первом запросе $l = 4$ и $k = 1$, $a = [1]$. На первом этапе удалится элемент 1, после этого $a = []$. Таким образом, за 1 этап удалился 1 элемент.
- Во втором запросе $l = 3$ и $k = 3$, $a = [2, 1]$. На первом этапе удалится элемент 2, после этого $a = [1]$. На втором этапе удалится элемент 1, после этого $a = []$. После третьего этапа массив a останется пустым. Таким образом, за 3 этапа удалилось 2 элемента.
- В третьем запросе $l = 1$ и $k = 3$, $a = [4, 3, 2, 1]$. На первом этапе удалится элемент 4, после этого $a = [3, 2, 1]$. На втором этапе удалится элемент 3, после этого $a = [2, 1]$. На третьем этапе удалится элемент 2, после этого $a = [1]$. Таким образом, за 3 этапа удалилось 3 элемента.

Задача F. Дорожный патруль

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

После открытия новой платной трассы M123 возникла проблема установки ограничения скорости. Так как на новую трассу ушло немало средств, было решено возместить часть затрат за счёт штрафов водителей, которые превышают ограничение скорости.

Мы можем выбрать целое неотрицательное число k для ограничения скорости, после которого происходит следующий процесс. Известно, что по трассе мимо патруля будут ехать n машин, причём патруль будет наблюдать их в порядке $1, 2, \ldots, n$. Скорость i-й машины в тот момент, в который её увидит патруль, будет равна a_i. Если $a_i > k$, то машина будет остановлена, и водитель заплатит штраф $a_i - k$. Но водители следующих t машин (т.е. с номерами $i + 1, i + 2, \ldots, i + t$) увидят, что машину остановил патруль, и они сбавят скорость так, что их уже не за что будет останавливать. Далее процесс будет повторяться аналогично.

Вам требуется выбрать такое k, чтобы максимизировать сумму денег, которую можно собрать через штрафы.

Формат входных данных

Первая строка содержит два целых числа n и t ($1 \leqslant t \leqslant n \leqslant 200\,000$) — общее количество машин и количество машин, которые снижают скорость непосредственно после каждой остановленной машины.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — скорости машин.

Формат выходных данных

Выведите одно целое число — максимальную суммарную величину штрафов, которую можно собрать при оптимальном ограничении скорости k.

Система оценки

Тесты к этой задаче состоят из примеров и 5 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Группы в данной задаче «склеиваются». Это означает, что за подгруппу начисляются баллы, если хотя бы одна из ваших посылок успешно проходит все её тесты. Обратите внимание, что если у подгруппы имеются необходимые подгруппы, то для проверки на данной подгруппе текущая посылка должна пройти тесты во всех этих необходимых подгруппах (при этом результаты предыдущих посылок не учитываются).

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	t		
0	0	–	–	–	Тесты из условия.
1	18	$n \leqslant 3000$	–	0	–
2	21	–	–	–	$a_i = i$
3	23	–	$t \geqslant 3000$	–	–
4	19	–	$t = 1$	–	–
5	19	–	–	0 – 4	–

Примеры

стандартный ввод	стандартный вывод
3 1 1 2 3	4
3 2 1 2 3	1
7 2 1 2 6 3 1 9 2	11
10 3 5 3 7 1 8 10 2 8 1 11	21

Замечание

В первом примере $k = 0$ (движение перекрыто), штраф заплатят первая и третья машины.

Во втором примере ответ 1 получается при любом из значений k от 0 до 2.

В третьем примере выгодно выбрать $k = 2$, тогда получится остановить третью и шестую машину, которые едут со скоростями 6 и 9, что позволит получить штрафы в сумме $(6 - 2) + (9 - 2) = 11$. Обратите внимание, что первые две машины мы не остановим, потому что их скорости не превышают ограничения.

В четвёртом примере оптимально сделать ограничение по скорости равным 1. В таком случае мы остановим первую, пятую и десятую машины и получим суммарный штраф $(5 - 1) + (8 - 1) + (11 - 1) = 21$.