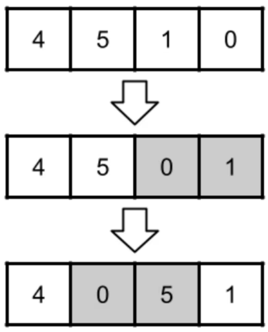
Задача A. Обмен

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

Дан массив A длины N, состоящий из целых чисел: $A = A_1, A_2, \ldots, A_N$. Разрешается выполнять обмен двух **соседних** элементов массива (то есть поменять местами A_i и A_{i+1} за один обмен).

Требуется определить **минимальное** количество таких обменов, необходимое для того, чтобы в получившемся массиве существовало **не более одной** пары соседних элементов разной чётности (один чётный, другой нечётный). Обратите внимание, что 0 также считается чётным числом.

Например, рассмотрим массив $A = [4, 5, 1, 0]$. Если последовательно выполнить обмен последних двух элементов и затем обмен средних двух элементов, то массив станет $[4, 0, 5, 1]$, и чётные и нечётные числа будут соседями не более одного раза.



Рисунок

Формат входных данных

В первой строке вводится одно число $1 \leqslant N \leqslant 10^5$ — количество элементов в массиве.

Во второй строке вводятся N чисел $0 \leqslant a_i \leqslant 2 \cdot 10^9$ — элементы массива.

Формат выходных данных

В единственной строке выведите минимальное количество обменов соседних элементов, необходимое для выполнения условия.

Пример

стандартный ввод	стандартный вывод
1 1058939805	0

Задача B. Железная дорога

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

У Леши есть железная дорога. Состав состоит из одного вагона и одного локомотива. На дороге есть N станций, пронумерованных последовательно $1, 2, \ldots, N$. Состав сейчас находится на станции 1.

Между каждой парой соседних станций установлены датчики: датчик между станциями i и $i+1$ считает, сколько раз локомотив проехал по этому участку, и сохраняет это число в облачное хранилище.

Пока Кирилл обедал, его друг Леша запускал поезд так, что в конце маршрута поезд оказался там же, откуда начал маршрут, — на станции 1. Поезд не может выезжать левее станции 1 и правее станции N (там тупики). Известно, что поезд может менять направление только на станциях, а расстояние между любыми соседними станциями он преодолевает за 1 секунду. Также известно, что поезд очень тяжёлый, поэтому Леша не мог поднимать его и переносить на другую станцию руками.

Кирилл решил выяснить, сколько существует возможных маршрутов поезда, которые состав мог совершить за время обеда. Однако у Кирилла есть только информация с датчиков $A_1, A_2, \ldots, A_{N-1}$, где A_i — сколько раз поезд проехал между станциями i и $i+1$.

Так как поезд тяжёлый, смена направления тоже задача непростая, поэтому Кирилл интересуют только маршруты с **минимальным** количеством смен направления (смена направления происходит, когда поезд после очередного переезда между станциями едет в противоположную сторону относительно предыдущего переезда). Необходимо посчитать количество таких маршрутов.

Гарантируется, что существует хотя бы один подходящий маршрут.

Формат входных данных

В первой строке вводится целое число M ($1 \leqslant M \leqslant 10^5$) — количество датчиков. Соответственно, число станций на 1 больше: $N = M + 1$.

Во второй строке вводятся M чисел через пробел $A_1, \ldots, A_M$ ($1 \leqslant A_i \leqslant 10^6$).

Формат выходных данных

Выведите ответ — количество подходящих маршрутов по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
2 6 4	3

Замечание

В первом примере все возможные маршруты с минимальным количеством смены направлений:
`RRLLRRLLRL`
`RLRRLLRRLL`
`RRLLRLRRLL`

Задача C. Списывание в школе

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

В школе ЦПМ планируется проведение контрольной работы!

В учебном классе этой школы n рядов одиночных парт по m штук в каждом. Известно, что некоторые ученики не смогут написать контрольную работу (разумеется, по уважительной причине).

Ученики школы ЦПМ ожидают, что эта контрольная будет очень сложной! Поэтому они *тщательно* готовились к ней, и сейчас ученик, сидящий за партой в i-м ряду на j-м месте, обладает бесконечным количеством шпаргалок (достаточно хорошо спрятанных от учителя) с эффективностью $s_{i,j}$ каждая. Но, как кажется, этого может не хватить.

Поэтому ученики придумали гениальный план помощи друг другу: ученик, сидящий за партой (i, j), может передать свои шпаргалки соседям по сторонам света в клетках $(i + 1, j)$, $(i - 1, j)$, $(i, j+1)$, $(i, j-1)$, если такие соседи существуют. Получив шпаргалки, ученик также может передать их дальше другим ученикам.

Однако есть *нюанс*. Во время контрольной учитель может удалить q учеников из класса за *«академическую нечестность»*. За одно удаление учитель выводит школьника из класса и забирает у всех остальных учеников все *чужие* шпаргалки (то есть после каждого удаления у каждого остаются только его собственные шпаргалки).

Ученики хотят действовать оптимально: в каждой текущей ситуации в классе они делятся шпаргалками так, чтобы эффективность шпаргалки у **каждого** оставшегося ученика была максимально возможной. Помогите им: найдите **минимальную** эффективность шпаргалки, которой может обладать ученик в классе, сразу после начала контрольной и после каждого из q удалений.

Гарантируется, что существует хотя бы один подходящий сценарий (то есть после всех удалений в классе остаётся хотя бы один ученик).

Формат входных данных

В первой строке входного файла вводятся два числа n и m — размеры класса ($1 \leqslant n \cdot m \leqslant 10^5$).

В следующих n строках вводятся по m целых чисел $s_{i,j}$ — эффективность шпаргалок ученика, сидящего за партой в ряду i на месте j. Если $s_{i,j} = 0$, то ученика за этой партой нет (он не придёт на контрольную). ($0 \leqslant s_{i,j} \leqslant 10^9$).

Далее вводится целое число q — количество учеников, которые будут удалены из класса по ходу работы ($0 \leqslant q \leqslant p - 1$), где p — количество занятых мест в классе в начале контрольной.

В следующих q строках вводятся пары чисел r_k, c_k — удаление ученика, сидящего за партой в ряду r_k на месте c_k ($1 \leqslant r_k \leqslant n$, $1 \leqslant c_k \leqslant m$). Гарантируется, что в момент удаления на месте (r_k, c_k) действительно сидит ученик.

Формат выходных данных

Выведите $q+1$ строку: минимальную эффективность шпаргалки, которой может обладать ученик класса в начале контрольной и после каждого из q удалений.

Пример

стандартный ввод	стандартный вывод
3 3 3 1 2 1 5 1 1 1 9 3 3 3 2 2 1 1	9 5 3 1

Замечание

Сначала ученик за третьей партой третьего ряда может помочь всем (в том числе опосредованно), поэтому минимальная эффективность шпаргалки у любого ученика в классе равна 9.

После удаления ученика за третьей партой третьего ряда школьник со шпаргалкой эффективности 5 за второй партой второго ряда всё так же может помочь всему классу. После его удаления ученик за первой партой первого ряда может поделиться с классом шпаргалками эффективности 3.

После удаления школьника за первой партой первого ряда класс «разбивается» на две части, которые никак не могут помочь друг другу. Поэтому минимальная по эффективности шпаргалка, которой может обладать ученик при оптимальной помощи одноклассников, имеет эффективность 1.

Задача D. Гонки

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

Данис имеет большую коллекцию гоночных машинок. Сегодня Данис показывает Марку свою коллекцию и для этого расставил машинки на плоскости. Каждая машинка задаётся отрезком длины 1, параллельным оси OX, и координатой своего левого конца. Например, запись $(-2, 5)$ означает машинку с концами $(-2, 5)$ и $(-1, 5)$.

Также у каждой машинки есть характеристика t — время, за которое она сдвигается на 1 вправо. Например, машинка из точки $(-2, 5)$ с характеристикой $t = 50$ через 50 единиц времени перейдёт в положение с концами $(-1, 5)$ и $(0, 5)$. При этом в любой момент времени между 0 и 50 эта машинка будет где-то на пути.

Марк ожидает появления машинок в точке $(0, 0)$, наблюдая вдоль оси OY. Марк **видит** машинку (на уровне $y = a$) тогда и только тогда, когда в координате $x = 0$ в некоторый момент времени находится часть (возможно, левый или правый конец) этой машинки, и при этом других машинок в точке $x = 0$ между $y = 0$ и $y = a$ нет (то есть ничто не блокирует Марку обзор).

Машинки стартуют одновременно из своих первоначальных позиций. Вычислите количество машинок, которые увидит Марк за всё время.

Формат входных данных

В первой строке вводится целое число N $(1 \leqslant N \leqslant 5 \cdot 10^4)$ — количество машинок Даниса.

Следующие N строк описывают машинки. В i-й строке заданы три целых числа x, y, t $(1 \leqslant y, t \leqslant 10^6,\ -1000 \leqslant x \leqslant -1)$. Все значения y различны.

Формат выходных данных

Выведите единственное целое число — количество машинок, которые увидит Марк.

Пример

стандартный ввод	стандартный вывод
3 -2 1 3 -3 2 3 -5 100 1	2

Замечание

В примере Марк увидит машинки 1 и 2, а 3 не увидит.

Задача E. Коровы

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

Коровы собираются построить *американские горки* и хотят сделать их как можно более весёлыми, но при этом уложиться в заданный бюджет.

Горки располагаются на прямой отрезке длины L (от точки 0 до точки L), где $1 \leqslant L \leqslant 1000$. У коров есть N различных *заготовок* (компонентов) для трассы. Компонент i имеет длину W_i $(1 \leqslant W_i \leqslant L)$ и из-за особенностей рельефа может начинаться только в точке X_i $(0 \leqslant X_i \leqslant L - W_i)$, то есть он покрывает отрезок $[X_i,\ X_i + W_i]$.

Также каждому компоненту сопоставлены:

- рейтинг веселья F_i $(1 \leqslant F_i \leqslant 10^6)$;
- стоимость C_i $(1 \leqslant C_i \leqslant 1000)$.

Коровы хотят выбрать несколько компонентов и выстроить из них непрерывную трассу, начиная в 0 и заканчивая в L, так что конец каждого выбранного компонента (кроме последнего) совпадает с началом следующего. Каждый компонент можно либо использовать, либо не использовать.

Суммарное веселье трассы равно сумме F_i по всем использованным компонентам, а суммарная стоимость — сумме C_i по всем использованным компонентам. Бюджет равен B $(1 \leqslant B \leqslant 1000)$.

Найдите максимальное возможное суммарное веселье трассы, которую можно построить в рамках бюджета. Если построить трассу от 0 до L в рамках бюджета невозможно, выведите -1.

Формат входных данных

В первой строке вводятся три числа L, N, B.

В следующих N строках задаются компоненты: по 4 числа X_i, W_i, F_i, C_i.

Формат выходных данных

В единственной строке выведите одно число — максимальный суммарный рейтинг веселья. Если решения нет, выведите -1.

Пример

стандартный ввод	стандартный вывод
5 6 10 0 2 20 6 2 3 5 6 0 1 2 1 1 1 1 3 1 2 5 4 3 2 10 2	17

Задача F. Ветер

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	8 секунд
Ограничение по памяти:	512 мегабайт

Игорь в подарок на день рождения получил перестановку a длины n.

Друг Игоря Иван задаёт ему q вопросов. Каждый вопрос задаётся двумя числами l и r. В ответ на вопрос Игорь должен сообщить количество последовательностей индексов $(t_1, t_2, \ldots, t_k)$ произвольной длины $k \geqslant 1$ таких, что:

- $l \leqslant t_i \leqslant r$ для каждого i от 1 до k;
- $t_i < t_{i+1}$ для каждого i от 1 до $k-1$;
- $a_{t_{i+1}}$ делится на a_{t_i} для каждого i от 1 до $k-1$.

Помогите Игорю и ответьте на все вопросы Ивана.

Формат входных данных

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

Первая строка каждого набора входных данных содержит два целых числа n и q ($1 \leqslant n, q \leqslant 10^6$) — длина перестановки и количество вопросов Ивана.

Вторая строка каждого набора входных данных содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$) — перестановка a.

В каждой из следующих q строк каждого набора входных данных содержатся два целых числа l и r ($1 \leqslant l \leqslant r \leqslant n$) — очередной вопрос Ивана.

Гарантируется, что сумма значений n и сумма значений q по всем наборам входных данных не превосходит 10^6.

Формат выходных данных

Для каждого набора входных данных выведите ответы на все вопросы Ивана.

Пример

стандартный ввод	стандартный вывод
4 8 8 2 1 6 3 5 4 8 7 1 8 2 8 1 7 1 6 1 3 5 8 4 4 2 3 1 1 1 1 1 3 3 3 2 1 1 2 1 3 2 3 8 1 1 2 3 4 5 6 7 8 1 8	20 15 18 12 5 5 1 3 1 2 3 2 27

Замечание

Все подходящие массивы в первом наборе входных данных: (1), (2), (3), (4), (5), (6), (7), (8), $(1,3)$, $(1,6)$, $(1,7)$, $(1,6,7)$, $(2,3)$, $(2,4)$, $(2,5)$, $(2,6)$, $(2,7)$, $(2,8)$, $(2,6,7)$, $(6,7)$.

Все подходящие массивы в четвертом наборе входных данных: (1), (2), (3), (4), (5), (6), (7), (8), $(1,2)$, $(1,3)$, $(1,4)$, $(1,5)$, $(1,6)$, $(1,7)$, $(1,8)$, $(1,2,4)$, $(1,2,6)$, $(1,2,8)$, $(1,2,4,8)$, $(1,3,6)$, $(1,4,8)$, $(2,4)$, $(2,6)$, $(2,8)$, $(2,4,8)$, $(3,6)$, $(4,8)$.

Задача G. Лестница

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

На раскопках нашли каменную террасу шириной ровно 1 метр, сложенную из кубиков $1 \times 1 \times 1$. Терраса представляет собой **ступенчатый профиль** из M ступеней, идущих снизу вверх. i-я ступень (нумерация снизу вверх) имеет:

- горизонтальную длину L_i (сколько метров она тянется по земле),
- вертикальный подъём H_i (на сколько метров уровень поднимается на этой ступени).

То есть, проходя по ступеням от первой к M-й, вы каждый раз идёте вперёд на L_i и поднимаетесь на H_i.

Чтобы сделать подъём удобнее, хотят уменьшить число ступеней до N ($N < M$). Разрешается **только добавлять** каменные кубики $1 \times 1 \times 1$ так, чтобы некоторые соседние ступени **слились** в одну более длинную ступень (промежуточные "перепады" высоты внутри объединённого участка должны исчезнуть за счёт достройки кубиками).

Высоты и длины ступеней новой лестницы могут отличаться от исходных. Найдите **минимальное** количество добавленных кубиков, достаточное для получения лестницы ровно из N ступеней.

Формат входных данных

В первой строке заданы два целых числа M и N ($1 \leqslant N < M \leqslant 100$).

Далее идут M строк, в каждой по два целых числа L и H — длина и подъём i-й ступени соответственно ($1 \leqslant L, H \leqslant 10^1$). Ступени нумеруются снизу вверх.

Формат выходных данных

Выведите одно число — минимальное количество кубиков $1 \times 1 \times 1$, которое нужно добавить.

Пример

стандартный ввод	стандартный вывод

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

Задача H. Переулок

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

В узком переулке размера $n \times m$ некоторые клетки заняты стенами, остальные свободны. Нужно протащить через переулок длинную *штангу* (прямой отрезок), которая всегда занимает ровно k соседних клеток в **одном столбце**.

Штангу разрешено держать только вдоль одной из вертикальных границ переулка:

- либо в левом столбце ($x = 1$),
- либо в правом столбце ($x = m$).

То есть в любой момент времени штанга занимает клетки $(r, 1), (r + 1, 1), \ldots, (r + k - 1, 1)$ или $(r, m), (r + 1, m), \ldots, (r + k - 1, m)$ для некоторого r.

Старт: штангу можно поставить в **любой** допустимой позиции в левом столбце (все k клеток должны быть свободны). Финиш: нужно оказаться в **какой-нибудь** допустимой позиции в правом столбце (все k клеток свободны).

Разрешены два типа действий:

1. **Сдвиг вдоль текущей границы** на одну клетку вверх или вниз (то есть изменить r на $r-1$ или $r + 1$), если после сдвига все занимаемые клетки свободны и находятся внутри поля.
2. **Перенос на другую границу** без изменения вертикального положения: если штанга занимала строки $r, \ldots, r+k-1$ у левой границы, то её можно переместить к правой границе на те же строки, и наоборот, **только если** все k целевых клеток свободны.

Определите максимальное k, для которого штангу можно пронести из левого столбца в правый. Если нельзя пронести штангу никакой длины, выведите 0.

Формат входных данных

В первой строке заданы два целых числа n и m ($1 \leqslant n, m \leqslant 300$).

В следующих n строках задано по m символов. Символ равен `#`, если в соответствующей клетке находится стена, иначе символ равен `.`.

Формат выходных данных

В единственной строке выведите одно число — максимальную длину k (в клетках), которую можно пронести. Если решения нет, выведите 0.

Пример

стандартный ввод	стандартный вывод
3 5 ...## .#.#. ##...	2

Задача I. Поездка на олимпиаду

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

Рассматриваются t независимых запросов. Для каждого запроса задан отрезок целых чисел $[l, r]$.

Для запроса требуется выбрать три **попарно различных** целых числа a, b, c такие, что $l \leqslant a, b, c \leqslant r$, и значение

$$(a \oplus b) + (b \oplus c) + (a \oplus c)$$

максимально возможно, где $\oplus$ — побитовое исключающее ИЛИ (XOR). Если оптимальных троек несколько, разрешается вывести любую.

Формат входных данных

В первой строке задано целое число t ($1 \leqslant t \leqslant 10^4$) — количество запросов.

В каждой из следующих t строк заданы два целых числа l и r ($0 \leqslant l < r < 2^{30}$, $r - l > 1$).

Формат выходных данных

Для каждого запроса выведите три попарно различных целых числа a, b, c ($l \leqslant a, b, c \leqslant r$), максимизирующие указанное выражение.

Пример

стандартный ввод	стандартный вывод
8 0 2 0 8 1 3 6 22 128 137 69 98 115 127 0 1073741823	1 2 0 8 7 1 2 1 3 7 16 11 134 132 137 98 85 76 123 121 118 965321865 375544086 12551794

Замечание

В первом наборе входных данных единственная, с точностью до перестановки, подходящая тройка чисел (a, b, c) это $(0, 1, 2)$.

Во втором наборе входных данных одна из подходящих троек это $(8, 7, 1)$, $(8 \oplus 7) + (7 \oplus 1) + (8 \oplus 1) = 15 + 6 + 9 = 30$, можно показать, что 30 — максимально возможное значение $(a \oplus b) + (b \oplus c) + (a \oplus c)$ при $0 \leqslant a, b, c \leqslant 8$.

Задача J. Забор

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

Рассмотрим забор из n вертикальных планок, пронумерованных от 1 до n слева направо. Каждой планке сопоставлен целочисленный цвет.

Необходимо обработать m операций двух типов:

- **Перекрашивание** ($q = 0$): заданы l, r, k. Требуется присвоить цвет k всем планкам с номерами от l до r включительно.
- **Сравнение** ($q = 1$): заданы l, r, k. Требуется определить, совпадают ли два фрагмента забора длины k, начинающиеся в позициях l и r соответственно, то есть равны ли

$$(c_l, c_{l+1}, \ldots, c_{l+k-1}) \quad \text{и} \quad (c_r, c_{r+1}, \ldots, c_{r+k-1}).$$

Для каждой операции сравнения нужно вывести символ: `+`, если фрагменты совпадают, и `-` иначе.

Формат входных данных

Первая строка содержит одно целое число n — количество планок ($1 \leqslant n \leqslant 100\,000$).

Вторая строка содержит n целых чисел $c_1, c_2, \ldots, c_n$ — начальные цвета планок.

Третья строка содержит одно целое число m — количество операций ($1 \leqslant m \leqslant 100\,000$).

Следующие m строк содержат описания операций: четыре целых числа q, l, r, k.

- Если $q = 0$, то это перекрашивание отрезка $[l, r]$ в цвет k ($1 \leqslant l \leqslant r \leqslant n$).
- Если $q = 1$, то это сравнение двух отрезков длины k, начинающихся в l и r ($1 \leqslant l, r \leqslant n - k + 1,\ k > 0$).

Все числа во входном файле положительны и не превосходят 100 000.

Формат выходных данных

Выведите одну строку: для каждой операции сравнения выведите `+` при совпадении соответствующих фрагментов и `-` иначе, в порядке их появления во входных данных.

Примеры

стандартный ввод	стандартный вывод
7 1 2 1 3 1 2 1 3 0 4 5 2 1 3 1 2 1 3 1 3	+-
2 1 2 5 1 1 2 1 0 2 2 1 1 1 2 1 0 1 2 3 1 1 1 2	-++

Задача K. Графы графы...

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

Даны два связных неориентированных графа G_1 и G_2 на одном и том же множестве вершин $\{1, 2, \ldots, n\}$. В каждом графе находится фишка: в G_1 она изначально стоит в вершине s_1, а в G_2 — в вершине s_2.

Бесконечное число раз выполняется следующая операция:

- Пусть перед операцией фишка в G_1 находится в вершине v_1, а фишка в G_2 — в вершине v_2.
- Выбирается вершина u_1, смежная с v_1 в графе G_1.
- Выбирается вершина u_2, смежная с v_2 в графе G_2.
- Фишки перемещаются: в G_1 из v_1 в u_1, в G_2 из v_2 в u_2.
- Стоимость этой операции равна $|u_1 - u_2|$.

Требуется определить минимально возможную суммарную стоимость всех операций (сумма по бесконечному числу шагов), либо сообщить, что эту сумму невозможно сделать конечной.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке задано целое число t $(1 \leqslant t \leqslant 500)$ — количество наборов.

Далее для каждого набора:

- В первой строке заданы n, s_1, s_2 $(2 \leqslant n \leqslant 1000,\ 1 \leqslant s_1, s_2 \leqslant n)$.
- Во второй строке задано m_1 $(1 \leqslant m_1 \leqslant 1000)$ — число рёбер в G_1.
- В следующих m_1 строках заданы рёбра G_1: по два числа a_i, b_i $(1 \leqslant a_i, b_i \leqslant n,\ a_i \neq b_i)$.
- Затем задано m_2 $(1 \leqslant m_2 \leqslant 1000)$ — число рёбер в G_2.
- В следующих m_2 строках заданы рёбра G_2: по два числа c_i, d_i $(1 \leqslant c_i, d_i \leqslant n,\ c_i \neq d_i)$.

Гарантируется, что оба графа связны. Также гарантируется, что суммы n, m_1 и m_2 по всем наборам входных данных не превосходят 1000.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число: минимальную возможную суммарную стоимость всех операций, или `-1`, если эту величину нельзя сделать конечной.

Пример

стандартный ввод	стандартный вывод
3 4 1 1 4 1 2 2 3 3 4 4 1 4 1 2 2 3 3 4 4 1 4 1 2 4 1 2 2 3 3 4 4 1 4 1 2 2 3 3 4 4 1 7 7 2 7 1 6 2 1 3 2 3 4 5 1 7 3 7 5 6 5 1 5 6 5 7 6 3 7 2 7 4	0 -1 7

Замечание

В одном из примеров можно построить бесконечную последовательность переходов по вершинам $2, 3, 4, 1, 2, 3, 4, 1, \ldots$, по которой фишка может двигаться и в первом, и во втором графе. В другом примере можно показать, что стоимость любой операции строго положительна, поэтому суммарная стоимость неизбежно будет бесконечной.

Задача L. Заказ мерча

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Дан массив $a_1, a_2, \dots, a_n$ — размеры n кофт, пронумерованных от 1 до n. Разрешается выбрать любой непустой подотрезок $[l, r]$.

Для подотрезка $[l, r]$ определим его удобство как

$$\mathrm{comfort}(l, r) = \max(a_l, a_{l+1}, \dots, a_r) - \min(a_l, a_{l+1}, \dots, a_r) - (r - l).$$

То есть это разброс значений на подотрезке минус количество промежутков между соседними элементами.

Требуется найти максимальное значение $\mathrm{comfort}(l, r)$ по всем парам $1 \leqslant l \leqslant r \leqslant n$. Далее выполняются q изменений: в i-м изменении элемент с индексом p заменяется на x. После каждого изменения (и также до всех изменений) требуется снова вывести максимум удобства.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке задано целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов.

Далее для каждого набора:

- В первой строке заданы n и q ($1 \leqslant n, q \leqslant 2 \cdot 10^5$).
- Во второй строке заданы n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leqslant a_i \leqslant 10^9$).
- В следующих q строках заданы изменения: два числа p и x ($1 \leqslant p \leqslant n,\ 1 \leqslant x \leqslant 10^9$), что означает присвоить $a_p := x$.

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

Формат выходных данных

Для каждого набора входных данных выведите $q + 1$ строку: максимальное значение удобства по всем подотрезкам до выполнения изменений и после каждого из q изменений.

Пример

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td><pre>3
2 2
1 10
1 10
2 2
5 3
1 2 3 4 5
3 7
1 4
5 2
8 5
7 4 2 4 8 2 1 4
5 4
1 10
3 2
8 11
7 7</pre></td><td><pre>8
0
7
0
4
4
4
5
3
6
6
9
7</pre></td></tr>
</table>

Замечание

Рассмотрим первый пример.

- До изменений можно взять весь массив, тогда $\max(a_1, a_2) - \min(a_1, a_2) - (2 - 1) = 10 - 1 - 1 = 8$.
- После первого изменения оба значения равны 10, оптимально взять отрезок длины 1, и удобство равно $10 - 10 - 0 = 0$.
- После второго изменения размеры равны 10 и 2, снова выгодно взять весь массив: $10 - 2 - 1 = 7$.