Задача A. Количество инверсий

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Напишите программу, которая для заданного массива $A = \langle a_1, a_2, \ldots, a_n \rangle$ находит количество пар (i, j) таких, что $i < j$ и $a_i > a_j$. Обратите внимание на то, что ответ может не влезать в int.

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leqslant n \leqslant 100\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов массива A — целых неотрицательных чисел, не превосходящих 10^9.

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 6 11 18 28 31	0
8 999994 999989 999982 999972 999969 999961 999954 999950	28

Задача B. Место встречи изменить нельзя

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

256 мегабайт

Даны N точек. Найдите такие две из них, что расстояние между ними минимально.

Формат входных данных

Первая строка входного файла содержит целое число N ($2 \leqslant N \leqslant 100\,000$) — количество точек. Каждая из следующих N строк содержит пару целых чисел X и Y, разделённых пробелом, — координаты ($-1\,000\,000\,000 \leqslant X, Y \leqslant 1\,000\,000\,000$). Все точки различны.

Формат выходных данных

Единственная строка выходного файла должна содержать координаты двух выбранных точек.

Пример

стандартный ввод	стандартный вывод
4 0 0 0 1 1 1 1 0	0 0 0 1

Задача C. Автобусные остановки

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

В деревне есть n домов, расположенных вдоль главной дороги, которую можно воспринимать как числовую прямую. i-й дом имеет координату x_i.

Жители предпосчитают автобусные остановки рядом с их домом, и чем дальше автобусная остановка, тем более несчастливы они. *Недовольство* дома оппределяется как квадрат расстояния между домом и ближайшей к нему автобусной остановкой. Ваша задача — построить k автобусных остановок вдоль главной дороги так, чтобы сумма недовольств домов была минимальна.

Обратите внимание: остановка может быть построена в любой точке числовой прямой, необязательно совпадающей с точкой какого-то из домов.

Формально, пусть ближайшая остановка к i-му дому находится в точке p_i. Тогда вы хотите минимизировать:

$$\sum_{i=1}^{n} |x_i - p_i|^2$$

Формат входных данных

Первая строка содержит два целых числа n, k ($1 \leqslant n \leqslant 5 \cdot 10^4$, $1 \leqslant k \leqslant \min(n, 100)$).

Вторая строка содержит n целых чисел x_i ($1 \leqslant x_i \leqslant 10^5$).

Формат выходных данных

Выведите одно число — ответ с относительной или абсолютной погрешностью не более 10^{-6}.

Пример

стандартный ввод	стандартный вывод
3 2 1 2 4	0.500000000000000

Замечание

Пусть построили автобусные остановки в координатах 1.5 и 4.0. Тогда:

- Недовольство первого дома: $(x_1 - p_1)^2 = (1.0 - 1.5)^2 = 0.25$
- Недовольство второго дома: $(x_2 - p_1)^2 = (2.0 - 1.5)^2 = 0.25$
- Недовольство третьего дома: $(x_3 - p_2)^2 = (4.0 - 4.0)^2 = 0.00$

Таким образом, суммарное недовольство равно $0.25 + 0.25 + 0.00 = 0.5$

Задача D. Прибавления на отрезках

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

Гриша пришел на контест и увидел там следующую задачу.

Дан массив длины n, изначально состоящий из нулей. Элементы массива пронумерованы от 1 до n. К массиву было применено q операций. i-я операция задается тремя целыми числами l_i, r_i и x_i $(1 \leqslant l_i \leqslant r_i \leqslant n)$, $(1 \leqslant x_i \leqslant n)$ и означает, что к элементам с номерами $l_i, l_i + 1, \ldots, r_i$ прибавили число x_i. Требуется найти максимум в массиве после применения всех этих операций.

Но Гриша не из глупых! Он решил эту задачу очень быстро.

Однако что-то в нем переклинило, и он задумался: «интересно, а какие значения может принять максимум в массиве после применения некоторого подмножества данных операций?».

Помогите Грише, найдите все такие целые числа y от 1 до n, что после применения некоторого (возможно, пустого) подмножества данных операций максимум в массиве равен y.

Формат входных данных

В первой строке находятся два целых числа n и q $(1 \leqslant n, q \leqslant 10^4)$ — длина массива и количество запросов в исходной задаче.

В следующих q строках описаны запросы, по одному в строке. i-я из этих строк содержит три целых числа l_i, r_i и x_i $(1 \leqslant l_i \leqslant r_i \leqslant n,\ 1 \leqslant x_i \leqslant n)$, что обозначает запрос на добавление числа x_i на отрезке с l_i-го по r_i-й элемент включительно.

Формат выходных данных

В первую строку выведите единственное число k, обозначающее количество возможных целых чисел от 1 до n, которым может быть равен максимум в массиве после применения некоторого (возможно, пустого) подмножества данных операций.

В следующей строке выведите через пробел все k чисел от 1 до n — возможные значения максимума. Выводите эти числа **в возрастающем порядке**.

Примеры

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

Замечание

Если в первом тестовом примере оставить только первый запрос, то максимум будет равен 1. Если оставить только второй запрос, то максимум будет равен 2. Если оставить первые два запроса, то максимум будет равен 3. Если оставить только третий запрос, то максимум будет равен 4. Но если оставить третий запрос и еще какой-то, максимум будет больше n, поэтому его выводить не требуется.

Во втором тестовом примере, оставив только первый запрос, можно получить 1. Оставив только второй, можно получить 2. А если оставить все запросы, максимум будет равен 3.

В третьем тестовом примере можно получить максимумы так:

- Можно получить максимум 2 оставив запросы: (1).
- Можно получить максимум 3 оставив запросы: (2).
- Можно получить максимум 5 оставив запросы: $(1, 2)$.
- Можно получить максимум 6 оставив запросы: (3).
- Можно получить максимум 8 оставив запросы: $(1, 3)$.
- Можно получить максимум 9 оставив запросы: $(2, 3)$.

Задача E. Большие Диаграммы Юнга

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

256 мегабайт

Яша и Саша — два роскошных брата, которые любят дарить друг другу подарки на день рождения. И вот, на очередной день рождения, Яша подарил Саше на день рождения Диаграмму Юнга. Саша решил посчитать число клеток в этой диаграмме. Но Саша опаздывал на матч по футболу, поэтому посчитал только число клеток в первой строке и первом столбце. И теперь ему интересно, какое максимальное по модулю число клеток может быть в этой диаграмме.

Формат входных данных

В первой строке входных данных находится одно целое число — размер первой строки диаграммы. Во второй строке находится другое целое число — размер первого столбца диаграммы. Длины этих чисел не превосходит двухсот пятидесяти тысяч символов. Размеры строк этой диаграммы могут быть отрицательными.

Формат выходных данных

В единственной строке выведите единственное число — максимальный по модулю размер диаграммы.

Пример

стандартный ввод	стандартный вывод
2 2	4

Задача F. Дружелюбные хомячки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

На плоскости живут n хомячков. Каждый в точке с целыми координатами. Хомячки дружат, если существует прямоугольник со сторонами, параллельными осям координат, содержащий этих двух хомячков и не содержащий никаких других.

Прямоугольник содержит хомячка, если точка, в которой он живет, лежит внутри прямоугольника или на его границе.

Сколько пар хомячков дружат?

Формат входных данных

На первой строке число n ($1 \leqslant n \leqslant 100\,000$).

Следующие n строк содержат по два целых числа – координаты точек, в которых живут хомячки.

Все точки различны, а координаты целые, по модулю не превосходят 10^9.

Формат выходных данных

Выведите одно целое число – количество пар дружащих хомячков.

Пример

стандартный ввод	стандартный вывод
5 0 0 0 2 2 0 2 2 1 1	8

Задача G. Очередная задача минимизации

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

Дан массив из n чисел $a_1 \dots a_n$. Стоимостью подотрезка элементов в массиве назовем количество неупорядоченных пар различных позиций внутри подотрезка, содержащих одинаковые элементы. Разбейте массив на k непересекающихся непустых подотрезков таких, что сумма их стоимостей минимальна. Каждый элемент массива должен попасть ровно в один подотрезок.

Формат входных данных

Первая строка содержит два целых числа n и k $(2 \leqslant n \leqslant 10^5,\ 2 \leqslant k \leqslant \min(n, 20))$ — размер массива и количество отрезков, на которые надо его разбить.

Следующая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ $(1 \leqslant a_i \leqslant n)$ — элементы массива.

Формат выходных данных

Выведите одно число — минимальную стоимость разбиения массива на подотрезки.

Примеры

стандартный ввод	стандартный вывод
7 3 1 1 3 3 3 2 1	1
10 2 1 2 1 2 1 2 1 2 1 2	8
13 3 1 2 2 2 1 2 1 1 1 2 2 1 1	9

Замечание

В первом примере оптимально разбить последовательность на три подпоследовательности: $[1]$, $[1, 3]$, $[3, 3, 2, 1]$. Стоимости равны 0, 0 и 1, поэтому ответ равен 1.

Во втором примере оптимально разбить подпоследовательность на две половины. Стоимость каждой половины равна 4.

В третьем примере оптимально разбить следующим образом: $[1, 2, 2, 2, 1]$, $[2, 1, 1, 1, 2]$, $[2, 1, 1]$. Стоимости равны 4, 4, 1.

Задача H. Ваня и бургеры

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

Ваня очень любит бургеры, а так же очень любит тратить деньги. На улице, где живет Ваня, находится n бургерных.

У Вани есть q друзей, i-й друг предложил Ване встретиться у бургерной с номером l_i и прогуляться до бургерной r_i ($l_i \leqslant r_i$). Во время прогулки с i-м другом Ваня может зайти во все бургерные с номерами x такими, что $l_i \leqslant x \leqslant r_i$.

Для каждой бургерной Ваня знает стоимость самого дорогого бургера c_i бурлей. Ваня хочет зайти в какое-то подмножество бургерных по пути, взять в каждой из них самый дорогой бургер и потратить как можно больше денег. Но есть небольшая проблема: карта Вани сломалась, и вместо того, чтобы снимать деньги после покупки, количество денег на карте меняется следующим образом.

Пусть до покупки у Вани было d бурлей и Ваня потратил c бурлей в бургерной, тогда после этих действий у Вани на счете останется $d \oplus c$ бурлей, где $\oplus$ обозначает операцию XOR.

Сейчас на счету Вани $2^{2^{100}} - 1$ бурлей и он уже собирается на прогулку. Помогите Ване узнать, какое максимальное количество денег он сможет потратить, если пойдет гулять с другом под номером i. Количество бурлей, потраченных Ваней, определяется как разность между начальным количеством бурлей на счете Вани и конечным количеством бурлей на его счете.

Формат входных данных

В первой строке находится одно целое число n ($1 \leqslant n \leqslant 500\,000$) — количество бургерных.

В следующей строке содержится n целых чисел $c_1, c_2, \ldots, c_n$ ($0 \leqslant c_i \leqslant 10^6$), где c_i — стоимость самого дорого бургера в бургерной под номером i.

В третьей строке находится одно целое число q ($1 \leqslant q \leqslant 500\,000$) — количество друзей Вани.

В каждой из следующих q строк находятся два целых числа l_i и r_i ($1 \leqslant l_i \leqslant r_i \leqslant n$) — пара номеров бургерных, между которыми совершается прогулка.

Формат выходных данных

Выведите q строк, в i-й строке выведите максимальное количество денег, которое Ваня может потратить вместе с другом под номером i.

Примеры

стандартный ввод	стандартный вывод
4 7 2 3 4 3 1 4 2 3 1 3	7 3 7
5 12 14 23 13 7 15 1 1 1 2 1 3 1 4 1 5 2 2 2 3 2 4 2 5 3 3 3 4 3 5 4 4 4 5 5 5	12 14 27 27 31 14 25 26 30 23 26 29 13 13 7

Замечание

В первом тесте для того, чтобы потратить максимальное количество денег с первым и третьим другом, Ване достаточно зайти в первую бургерную. Со вторым другом Ване достаточно зайти в третью бургерную. Во втором тесте для третьего друга (который собирается прогуляться от первой до третьей бургерной) всего есть 8 вариантов потратить деньги — 0, 12, 14, 23, $12 \oplus 14 = 2$, $14 \oplus 23 = 25$, $12 \oplus 23 = 27$, $12 \oplus 14 \oplus 23 = 20$. Максимальное количество денег получается потратить, если зайти в первую и третью бургерную — $12 \oplus 23 = 27$.