

Общая информация по задачам пробного тура

Задача	Тип задачи	Ограничения
X. Сумма двух	Стандартная	1 с, 512 МБ
Y. Разворот последовательности	Стандартная	2 с, 512 МБ
Z. Угадай число	Интерактивная	1 с, 512 МБ

Необходимо считывать данные из стандартного потока ввода. Выходные данные необходимо выводить в стандартный поток вывода.

В задачах X и Z, баллы за подзадачу начисляются только, если все тесты для подзадачи пройдены.

Задача X. Сумма двух

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 МБ

Заданы два целых числа.

Требуется написать программу, вычисляющую их сумму.

Формат входных данных

Входные данные содержат два целых числа a и b , по одному в строке ($-10^{100} \leq a, b \leq 10^{100}$).

Формат выходных данных

Выведите сумму заданных чисел.

Примеры

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

Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Результаты во время тура
		a, b		
1	25	$0 \leq a, b \leq 10^9$	—	потестовые
2	25	$-10^9 \leq a, b \leq 10^9$	У, 1	потестовые
3	25	$0 \leq a, b \leq 10^{100}$	1	только баллы
4	25	$-10^{100} \leq a, b \leq 10^{100}$	У, 1-3	только баллы

Задача Y. Разворот последовательности

Ограничение по времени: 2 секунды

Ограничение по памяти: 512 МБ

Задана последовательность целых чисел.

Требуется написать программу, которая выводит эту последовательность в обратном порядке.

Формат входных данных

Входные данные содержат последовательность целых чисел: a_1, a_2, \dots, a_n ($-10^{18} \leq a_i \leq 10^{18}$), соседние числа разделены одним пробелом. Количество элементов последовательности не превышает 10^6 .

Формат выходных данных

Выведите заданную во входных данных последовательность чисел в обратном порядке.

Пример

стандартный ввод	стандартный вывод
1 2 3	3 2 1

Система оценки

Для оценивания этой задачи используется 10 тестов. Каждый тест оценивается в 10 баллов. Тесты оцениваются независимо. По итогам проверки для каждого теста сообщается результат проверки на этом тесте.

Задача Z. Угадай число

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 МБ

Это интерактивная задача. В процессе тестирования программа участника взаимодействует с программой жюри с использованием стандартных потоков ввода/вывода.

Программа жюри загадала число от 1 до n , цель программы участника — отгадать, задав не более чем за 30 вопросов. Для этого программа участника сообщает свои догадки программе жюри, а программа жюри отвечает, является ли загаданное число бóльшим, меньшим или равным сделанной догадке.

Протокол взаимодействия

Сначала необходимо прочитать из стандартного потока ввода число n ($1 \leq n \leq 10^9$). Затем протокол общения следующий: требуется вывести в стандартный поток вывода одну строку, содержащую целое число — свою догадку о загаданном числе.

После этого необходимо считать из стандартного потока ввода одно число: сообщение программы жюри. Возможны следующие сообщения:

- «1» — загаданное число больше последней догадки;
- «-1» — загаданное число меньше последней догадки;
- «0» — последняя догадка верна. Считав 0, ваша программа должна завершиться.

Пример

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

Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Результаты во время тура
		n		
1	50	$1 \leq n \leq 30$	У	первая ошибка
2	50	$1 \leq n \leq 10^9$	У, 1	первая ошибка

Общая информация по задачам первого тура

Задача	Тип задачи	Ограничения
1. Постановочное фото	Стандартная	2 с, 512 МБ
2. Беспилотное такси	Стандартная	3 с, 512 МБ
3. Экспресс 20/19	Стандартная	3 с, 1024 МБ
4. Чёрная дыра	Интерактивная	1 с, 1024 МБ

Необходимо считывать данные из стандартного потока ввода. Выходные данные необходимо выводить в стандартный поток вывода.

Во всех задачах баллы за подзадачу начисляются, только если все тесты для подзадачи пройдены.

Задача 1. Постановочное фото

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Перед общим фотографированием участников всероссийской олимпиады школьников по информатике главный фотограф решил сделать постановочное фото для своих подписчиков в социальной сети *InnoGram*.

В олимпиаде принимают участие школьники из n регионов, каждая делегация состоит из m школьников. Делегация каждого региона хочет подчеркнуть свою индивидуальность, поэтому надела фирменные футболки своего цвета, который не совпадает с цветом футболок никакого другого региона. Обозначим цвет футболки, который надели школьники i -го региона, числом i .

Для организации постановочного фото фотограф планирует действовать следующим образом. На сцене в ряд расположены места, куда могут вставать школьники, они пронумерованы вдоль сцены от 1 до m . Фотограф планирует по очереди обратиться к руководителям некоторых делегаций с просьбой нескольким школьникам этой делегации выйти на сцену. При этом он указывает два числа: L и R . Школьники выбранной делегации выходят на сцену и занимают все места от L -го до R -го, включительно. Если на каких-либо из этих мест уже стоят школьники других делегаций, то они уходят со сцены, а их места занимают школьники новой делегации. Фотограф может обратиться к руководителю каждой делегации не более одного раза.

Для цветовой гармонии на получившемся снимке фотограф хочет, чтобы на фотографии стояли m школьников, причём цвета надетых на них футболок должны следовать в строго определенном порядке. Теперь он хочет понять, каким образом он может получить желаемую фотографию.

Требуется написать программу, которая по заданному порядку цветов футболок на фотографии определяет, в каком порядке следует попросить руководителей делегаций отправить школьников на сцену, и какие места им следует занять, чтобы сделать желаемое фото, либо выясняет, что это невозможно.

Формат входных данных

Первая строка входных данных содержит два целых числа: m и n ($1 \leq m \leq 3 \cdot 10^5$, $1 \leq n \leq 3 \cdot 10^5$). Вторая строка содержит m целых чисел a_1, a_2, \dots, a_m ($1 \leq a_i \leq n$) — цвета футболок в том порядке, в котором фотограф хочет получить их на фотографии.

Формат выходных данных

Первая строка выходных данных должна содержать одно целое число k . Если сделать желаемое фото невозможно, это число должно быть равно -1 . В противном случае оно должно быть равно количеству делегаций, к руководителям которых фотограф должен обратиться, чтобы сделать желаемое фото.

В этом случае следующие k строк должны описывать просьбы фотографа в том порядке, в котором их следует сделать. Его i -я просьба задается тремя целыми числами: c_i , L_i и R_i , где c_i — номер делегации, к которой следует обратиться, L_i и R_i — номера первого и последнего места на сцене, соответственно, которые необходимо занять школьникам делегации c_i ($1 \leq c_i \leq n$, все c_i должны быть различны, $1 \leq L_i \leq R_i \leq m$).

Если существует несколько решений, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
7 10 10 5 5 10 4 2 4	5 4 1 7 7 2 4 10 1 4 5 2 3 2 6 6
5 2 1 2 1 2 1	-1

Система оценки

Подз.	Баллы	Ограничения		Необх. подзадачи	Результаты во время тура
		m	n		
1	15	$m \leq 100$	$n \leq 100$	У	потестовые
2	15	$m \leq 10^4$	$n \leq 10^4$	У, 1	потестовые
3	5	$m \leq 3 \cdot 10^5$	$n \leq 2$		потестовые
4	5	$m \leq 3 \cdot 10^5$	$n \leq 3$	3	потестовые
5	20	$m \leq 3 \cdot 10^5$	$n \leq 10$	У, 3, 4	потестовые
6	40	$m \leq 3 \cdot 10^5$	$n \leq 3 \cdot 10^5$	У, 1–5	потестовые

Задача 2. Беспилотное такси

Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

На главной площади Иннополиса планируется демонстрация возможностей беспилотных такси. Площадь имеет форму прямоугольника размером $n \times m$ метров, разделённого на единичные квадраты. Строки прямоугольника пронумерованы от 1 до n , а столбцы — от 1 до m . Таким образом, каждый квадрат характеризуется двумя положительными числами r и c — номерами строки и столбца, на пересечении которых он находится. При движении по площади беспилотное такси перемещается между единичными квадратами, за один *шаг* оно может переместиться из квадрата, в котором оно сейчас находится, в квадрат, имеющий с ним общую сторону.

Зима в этом году выдалась снежная, и для уборки площади в процессе демонстрации планируется использовать беспилотные снегоуборочные машины. При этом система управления беспилотным транспортом находится в режиме тестирования, поэтому одновременно на площади может находиться только одно беспилотное транспортное средство.

Поскольку снег продолжает идти в течение всей демонстрации, инженеры беспилотного такси решили продемонстрировать его возможности по преодолению сугробов. В каждый момент времени *глубина снега* на каждом единичном квадрате площади характеризуется неотрицательным целым числом. *Проходимость* беспилотного такси также задаётся некоторым неотрицательным целым числом. При перемещении по площади такси может находиться только на тех единичных квадратах, глубина снега на которых не превышает его проходимости. Перед каждой поездкой инженеры могут настроить такси, задав при этом значение его проходимости.

Исходно глубина снега на всей площади равна нулю. В начале каждого часа глубина снега на каждом квадрате увеличивается на один. После этого система управления беспилотным транспортом может выполнить одно из трех действий, каждое из которых занимает ровно один час:

- 1) запустить снегоуборочную машину вдоль строки, после этого на всех квадратах этой строки глубина снега становится равной нулю;
- 2) запустить снегоуборочную машину вдоль столбца, после этого на всех квадратах этого столбца глубина снега становится равной нулю;
- 3) продемонстрировать возможности беспилотного такси: задать значение его проходимости, выбрать стартовый и финишный единичные квадраты, и попытаться проехать на такси от стартового квадрата до финишного.

Поездка от стартового до финишного квадрата возможна, если существует последовательность единичных квадратов, начинающаяся в стартовом единичном квадрате и завершающаяся в финишном, в которой любые два соседних квадрата имеют общую сторону, и глубина снега на каждом из квадратов последовательности не превышает проходимости такси. Если выставленное значение проходимости позволяет осуществить поездку, такси должно использовать путь с минимальным количеством шагов.

Для каждой поездки на беспилотном такси требуется определить, возможно ли осуществить поездку, и если да, то какое минимальное количество шагов потребуется для осуществления этой поездки.

Формат входных данных

Первая строка ввода содержит три целых числа n , m и q — размеры площади и количество действий, которые совершаются системой управления беспилотными такси ($1 \leq n, m \leq 10^6$, $1 \leq q \leq 300\,000$).

Следующие q строк содержат описания действий, i -я из них имеет один из следующих трех типов:

- «1 r_i » — запустить снегоуборочную машину вдоль строки r_i ($1 \leq r_i \leq n$).
- «2 c_i » — запустить снегоуборочную машину вдоль столбца c_i ($1 \leq c_i \leq m$).

- «3 $r_{i,1}$ $c_{i,1}$ $r_{i,2}$ $c_{i,2}$ k_i » — задать для такси значение проходимости, равное k_i , и попытаться проехать от единичного квадрата $(r_{i,1}, c_{i,1})$ до единичного квадрата $(r_{i,2}, c_{i,2})$ ($1 \leq r_{i,1}, r_{i,2} \leq n$, $1 \leq c_{i,1}, c_{i,2} \leq m$, $0 \leq k_i \leq q$, квадраты $(r_{i,1}, c_{i,1})$ и $(r_{i,2}, c_{i,2})$ различны).

Гарантируется, что во входных данных есть хотя бы одно действие типа 3.

Формат выходных данных

Для каждого действия типа 3 требуется вывести отдельную строку. Если поездка от стартового до финишного квадрата возможна, необходимо вывести минимальное количество шагов, необходимое для осуществления поездки. Если поездка невозможна, необходимо вывести число -1 .

Система оценки

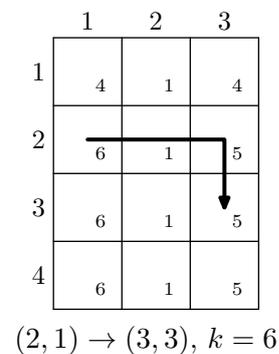
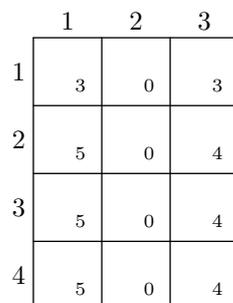
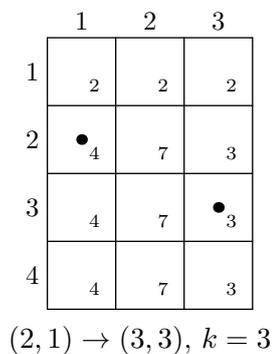
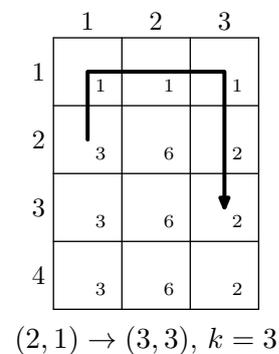
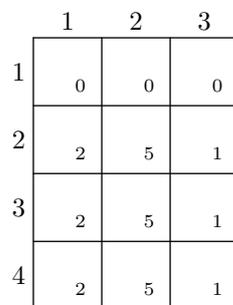
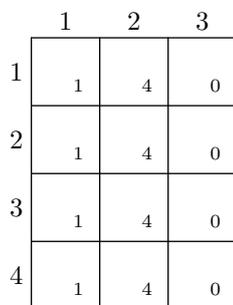
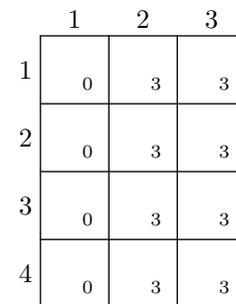
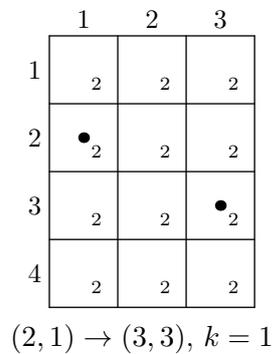
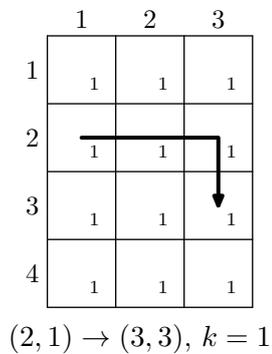
Подз.	Баллы	Ограничения			Необх. подзадачи	Результаты во время тура
		n, m	q	дополнительно		
1	19	$n, m \leq 50$	$q \leq 4000$		У	потестовые
2	20	$n, m \leq 10^4$	$q \leq 10^4$		У, 1	потестовые
3	19	$n, m \leq 10^6$	$q \leq 10^4$		У, 1, 2	потестовые
4	10	$n, m \leq 10^5$	$q \leq 10^5$		У, 1, 2	первая ошибка
5	10	$n, m \leq 10^6$	$q \leq 3 \cdot 10^5$	нет запросов типа 2	—	первая ошибка
6	11	$n, m \leq 10^6$	$q \leq 3 \cdot 10^5$	во всех запросах типа 3 значения k_i одинаковые	—	первая ошибка
7	11	$n, m \leq 10^6$	$q \leq 3 \cdot 10^5$		У, 1–6	только баллы

Пример

стандартный ввод	стандартный вывод
4 3 9	3
3 2 1 3 3 1	-1
3 2 1 3 3 1	5
2 1	-1
2 3	3
1 1	
3 2 1 3 3 3	
3 2 1 3 3 3	
2 2	
3 2 1 3 3 6	

Пояснение к примеру

На рисунке приведены уровни снега на площади перед каждым действием в примере и оптимальный маршрут такси для каждой успешной попытки проехать от одного квадрата до другого.



Задача 3. Экспресс 20/19

Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Система, связывающая столицу Флатландии с соседними городами с помощью электричек, сильно устарела. Для её модернизации было принято решение пустить из столицы до одной из станций электричку-экспресс.

Всего в железнодорожной сети Флатландии n станций, пронумерованных целыми числами от 1 до n , сама столица имеет номер 1. Всего существует m односторонних перегонов между станциями. По каждому из них экспресс может проехать от станции с меньшим номером до станции с большим номером. Для каждого перегона известно время, за которое экспресс проезжает этот перегон.

Маршрутом будем называть некоторую последовательность перегонов, такую, что первый перегон начинается в столице, и для любых двух последовательных перегонов начало второго совпадает с концом первого. Время проезда по маршруту равно суммарному времени прохождения всех этих перегонов, временем стоянки на станциях следует пренебречь.

Министерство транспорта Флатландии планирует рассмотреть несколько вариантов маршрута экспресса от столицы. Каждый вариант характеризуется номером станции, до которой следует пустить экспресс, и ориентировочным временем проезда по маршруту. При этом, в министерстве понимают, что точно выполнить требование о времени проезда может оказаться невозможно. Поэтому они используют для оценки допустимости маршрута величину p : маршрут с временем проезда x является допустимым для ориентировочного времени r , если $r \leq x \leq \frac{p}{p-1} \cdot r$.

Требуется написать программу, которая по описанию железнодорожной сети Флатландии и вариантам маршрута определяет для каждого варианта, существует ли допустимый для этого варианта маршрут экспресса.

Формат входных данных

В этой задаче один запуск программы должен обрабатывать несколько тестов.

В первой строке ввода находится единственное целое число t — количество тестов ($1 \leq t \leq 1000$). В следующих строках следует описание этих тестов друг за другом в следующем формате.

В первой строке находятся четыре целых числа n , m , q , p — количество станций, количество перегонов, количество вариантов, которые требуется рассмотреть, и параметр, задающий допустимое превышение ориентировочного времени проезда маршрута ($2 \leq n \leq 500\,000$, $1 \leq m \leq 500\,000$, $1 \leq q \leq 500\,000$, $2 \leq p \leq 20$).

В следующих m строках находятся по три целых числа, описывающих i -й перегон: v_i , u_i , d_i — станция отправления, станция прибытия и время, за которое экспресс проезжает этот перегон ($1 \leq v_i < u_i \leq n$, $1 \leq d_i \leq 10^{11}$). Гарантируется, что для любой станции существует хотя бы один маршрут из столицы, заканчивающийся в ней.

В следующих q строках находятся по два целых числа f_i и r_i — требование проверить существование допустимого маршрута до станции f_i с ориентировочным временем проезда r_i ($2 \leq f_i \leq n$, $1 \leq r_i \leq 10^{17}$).

Гарантируется, что каждая из сумм значений n , значений m и значений q по всем тестам одних входных данных не превосходит 500 000.

Формат выходных данных

Выведите t строк, по одной для каждого заданного во входных данных теста.

В качестве ответа на тест выведите строку s длины q , состоящую из символов 0 и 1. Символ s_i должен быть равен 1, если для i -го варианта существует допустимый маршрут, то есть маршрут до станции f_i , время проезда по которому лежит в отрезке $[r_i, \frac{p}{p-1} \cdot r_i]$, и 0 — в противном случае.

Примеры

стандартный ввод	стандартный вывод	иллюстрация
2 3 3 5 20 1 2 20 2 3 1 1 3 10 2 19 2 20 3 20 3 21 3 9 7 10 5 5 1 2 15 1 3 10 2 4 21 3 4 30 2 5 14 3 5 31 4 6 3 5 6 14 1 7 39 5 7 13 7 42 7 43 7 44 5 39 6 44	11110 10111	
1 4 6 5 2 1 2 1 2 3 1 3 4 1 1 2 70 2 3 120 3 4 4 4 90 4 2 4 10 4 37 2 34	11010	

Замечание

Во втором примере:

- В первом запросе подойдёт маршрут с временем $1 + 120 + 1 = 122$.
- Во втором запросе подойдёт маршрут с временем $1 + 1 + 1 = 3$.
- В четвёртом запросе подойдёт маршрут с временем $70 + 1 + 1 = 72$.

Для третьего и пятого запроса нет ни одного подходящего маршрута.

Система оценки

Подз.	Баллы	Ограничения		Необх. подзадачи	Результаты во время тура
		n, m, q	дополнительно		
1	15	$n \leq 10$ $m \leq 10$ $q \leq 10$		У	полные
2	24	$\sum n \leq 5000$ $\sum m \leq 5000$ $\sum q \leq 5000$	$r_i \leq 5000$	У	первая ошибка
3	17	$m = 2n - 2$ $q \leq 10$	$p = 2$, каждый перегон соединяет станции с номерами i и $i + 1$, причем из вершины i в вершину $i + 1$ проведено ровно два перегона для всех $1 \leq i \leq n - 1$	—	первая ошибка
4	11	$m = 2n - 2$	Каждый перегон соединяет станции с номерами i и $i + 1$, причем из вершины i в вершину $i + 1$ проведено ровно два перегона для всех $1 \leq i \leq n - 1$	3	только баллы
5	11	$\sum n \leq 1000$ $\sum m \leq 2000$	Все r_i равны	—	только баллы
6	11		Все r_i равны	5	только баллы
7	11			У, 1–6	только баллы

Задача 4. Чёрная дыра

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Это интерактивная задача.

Учёные планируют провести измерение уровней излучения ряда чёрных дыр. Уровень излучения чёрной дыры задаётся целым числом от 1 до n . Для измерения уровня излучения каждой чёрной дыры используется свой специальный орбитальный зонд, на котором установлен датчик излучения.

Датчик, установленный на зонде, может отвечать на следующие запросы: по значению x определить, верно ли, что уровень излучения больше или равен x . К сожалению, из-за ошибки в программном обеспечении ответ датчика может быть неверным. К счастью, после первого же неверного ответа датчик этого зонда изменяет своё состояние и на все последующие запросы выдаёт только верные ответы.

Учёные хотят выяснить уровень излучения нескольких чёрных дыр, выполнив для каждой из них не слишком много запросов к датчику направленного к ней зонда.

Требуется написать программу, которая взаимодействует с программой жюри, симулирующей датчики зондов, и определяет уровень излучения каждой чёрной дыры.

Для каждого запуска программы-решения необходимо решить задачу для нескольких чёрных дыр с одинаковым значением n . Количество чёрных дыр в одном запуске не превосходит 100 и не сообщается программе-решению.

Для каждого теста жюри зафиксировано число q — максимальное количество запросов, которые разрешается сделать для одной чёрной дыры. Гарантируется, что q запросов достаточно, чтобы решить задачу независимо от того, какие ответы будет давать программа жюри. Это число не сообщается программе-решению. Ограничения q в различных подзадачах приведены в таблице с информацией о системе оценки. Если программа-решение делает более q запросов к программе жюри для одной чёрной дыры, то на этом тесте она получает в качестве результата тестирования «Неверный ответ».

Протокол взаимодействия

Сначала на вход подаётся число n — максимальный возможный уровень излучения чёрной дыры ($1 \leq n \leq 30\,000$). Это число одинаковое для всех исследуемых в этом запуске чёрных дыр. После считывания n программа-решение должна интерактивно взаимодействовать с программой жюри, чтобы последовательно определить уровень излучения одной или нескольких чёрных дыр.

Программа должна выполнять запросы к датчику и получать ответы об уровне излучения.

Чтобы сделать запрос, программа-решение должна вывести строку вида «? x », где x — целое положительное число ($1 \leq x \leq n$).

В ответ на запрос программы-решения программа жюри подаёт ей на вход либо строку «Yes», если датчик сообщает, что уровень излучения чёрной дыры больше или равен значению x , заданному в запросе, либо строку «No», если датчик сообщает, что уровень излучения чёрной дыры меньше x .

Если программа-решение определила ответ на задачу, то она должна вывести строку «! x », где x — искомый уровень излучения чёрной дыры. Если x является единственным уровнем излучения, который противоречит не более чем одному ответу программы жюри для данной чёрной дыры, то ответ для этой чёрной дыры считается правильным.

После вывода ответа для очередной чёрной дыры программа должна считать строку. В случае, когда необходимо приступить к исследованию следующей чёрной дыры, на ввод будет подана строка «Correct».

Если все необходимые чёрные дыры были исследованы, то на ввод будет подана строка «Done». В этом случае программа-решение должна завершить работу.

Если для очередной чёрной дыры дан неправильный ответ, программа-решение будет принудительно завершена и на этом тесте получит результат тестирования «Неверный ответ».

Гарантируется, что после каждого ответа программы жюри существует такое значение уровня излучения, что ответы на все ранее сделанные запросы для данной чёрной дыры кроме, может быть, одного, верны. Программа жюри может адаптировать своё поведение, в том числе выбирать, какой

из уже данных ответов был неверным и чему равен уровень излучения исследуемой чёрной дыры, в зависимости от запросов, сделанных программой-решением.

Примеры

стандартный ввод	стандартный вывод
2	? 2
Yes	? 2
No	? 2
Yes	! 2
Correct	? 2
No	? 2
No	! 1
Done	
3	? 2
Yes	? 2
Yes	? 3
No	? 3
Yes	? 3
No	! 2
Done	

Пояснение к примеру

В первом примере для первой чёрной дыры после первых двух запросов неизвестно, на какой из них ответ датчика оказался неверным, поэтому необходим третий запрос.

Во втором примере показан один из возможных вариантов взаимодействия для $n = 3$, который находит ответ за 5 запросов. Можно показать, что за 4 запроса узнать ответ нельзя.

В обоих примерах решение проверяется со значением $q = 30$.

Обратите внимание, что программа жюри может давать другие ответы, даже если решение будет задавать ровно те же запросы, что в примере.

Система оценки

В подзадачах 3–17 в каждом тесте значение q равно минимальному числу разрешённых запросов, с помощью которых можно гарантированно (то есть сделав не более q запросов) узнать уровень излучения чёрной дыры независимо от ответов датчика для значения n в этом тесте.

Подз.	Баллы	Ограничения		Необх. подзадачи	Результаты во время тура
		n	q		
1	7	$n \leq 1000$	$q = 30$	У	потестовые
2	8	$n \leq 1000$	$q = 21$	У, 1	потестовые
3	6	$n \leq 4$	q оптимально	У	потестовые
4	9	$n \leq 7$	q оптимально	У, 3	потестовые
5	5	$n \leq 12$	q оптимально	У, 3–4	потестовые
6	6	$n \leq 25$	q оптимально	У, 3–5	потестовые
7	4	$n \leq 40$	q оптимально	У, 3–6	потестовые
8	5	$n \leq 80$	q оптимально	У, 3–7	первая ошибка
9	5	$n \leq 150$	q оптимально	У, 3–8	первая ошибка
10	8	$n \leq 300$	q оптимально	У, 3–9	первая ошибка
11	7	$n \leq 500$	q оптимально	У, 3–10	первая ошибка
12	5	$n \leq 1000$	q оптимально	У, 1–11	первая ошибка
13	5	$n \leq 2000$	q оптимально	У, 1–12	первая ошибка
14	5	$n \leq 4000$	q оптимально	У, 1–13	первая ошибка
15	5	$n \leq 8000$	q оптимально	У, 1–14	первая ошибка
16	5	$n \leq 15\,000$	q оптимально	У, 1–15	первая ошибка
17	5	$n \leq 30\,000$	q оптимально	У, 1–16	первая ошибка

Общая информация по задачам второго тура

Задача	Тип задачи	Ограничения
5. Гирлянда	Стандартная	2 с, 512 МБ
6. Классные парты	Стандартная	2 с, 512 МБ
7. Робогольф	Стандартная	3 с, 512 МБ
8. Поиск идеи	Стандартная	5 с, 1024 МБ

Необходимо считывать данные из стандартного потока ввода. Выходные данные необходимо выводить в стандартный поток вывода.

Во всех задачах баллы за подзадачу начисляются, только если все тесты для подзадачи пройдены.

Задача 5. Гирлянда

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Гирлянда — это цепочка из элементов: флажков и шариков, содержащая хотя бы один флажок. Длинной гирлянды называется количество элементов, из которых она состоит.

Гирлянда считается красивой, если для каждого флажка количество подряд идущих шариков слева от него до ближайшего слева флажка или до начала гирлянды равно количеству подряд идущих шариков справа от него до ближайшего справа флажка или до конца гирлянды.

Обозначим шарик символом «0», а флажок — символом «1». Например, гирлянда «0001000» является красивой, а гирлянда «001010» — нет, поскольку слева от первого флажка два шарика, а справа от него — один шарик. Отметим, что цепочка «000» не является гирляндой, так как не содержит ни одного флажка.

Задана гирлянда, разрешается удалить некоторые её элементы.

Требуется написать программу, находящую по описанию гирлянды самую длинную красивую гирлянду, которую можно получить из заданной удалением элементов.

Формат входных данных

Первая строка содержит целое число n — количество элементов в исходной гирлянде ($1 \leq n \leq 500\,000$).

Вторая строка содержит описание гирлянды в виде строки из n символов «0» и «1». Строка содержит хотя бы один символ «1».

Формат выходных данных

В первой строке выведите число m — длину получившейся красивой гирлянды ($1 \leq m \leq n$).

Во второй строке выведите получившуюся красивую гирлянду.

Если существует несколько решений, выведите любое из них.

Система оценки

Обозначим через k количество флажков в исходной гирлянде.

Подз.	Баллы	Ограничения		Необх. подзадачи	Результаты во время тура
		n	k		
1	20	$n \leq 15$		У	потестовые
2	20	$n \leq 1000$	$k \leq 2$	—	потестовые
3	20	$n \leq 1000$	$k \leq 15$	У, 1, 2	потестовые
4	16	$n \leq 1000$		У, 1–3	потестовые
5	10	$n \leq 100\,000$	$k \leq 50$	У, 1–3	первая ошибка
6	14	$n \leq 500\,000$		У, 1–5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
10 0100100000	7 0001000
3 111	3 111
7 0100101	5 01010

Задача 6. Классные парты

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Для нового кабинета школы Иннополиса требуется купить n двухместных парт.

Парты бывают k типов, которые задают их размер. Парты типа i подходят школьникам, рост которых находится в диапазоне от L_i до R_i включительно. Остальным школьникам сидеть за такой партой неудобно, при этом *величиной неудобства* школьника, если он сидит за такой партой, будем называть модуль разности его роста и ближайшей границы диапазона этой парты. Если парты школьнику подходит, то для него величина неудобства равна нулю.

Например, если $L_i = 100$ и $R_i = 120$, то неудобство для школьника с ростом 80 равно 20, для школьника с ростом 130 равно 10, а для школьника с ростом 105 равно 0.

В кабинете по очереди будут заниматься m групп школьников, каждая из которых состоит из $2n$ человек. Известен рост каждого школьника в каждой из групп. Закупленные парты будут расставлены в классе, и в каждой группе за каждой партой будут сидеть ровно два школьника. Необходимо купить n парт и рассадить за ними школьников каждой группы таким образом, чтобы суммарное неудобство для всех школьников, занимающихся в этом кабинете, было минимальным.

Требуется написать программу, которая по информации о каждом из k типов парт и известным значениям роста каждого школьника в каждой группе определяет, какого минимального суммарного значения неудобства школьников можно достичь, купив парты и рассадив за них школьников в каждой группе оптимальным образом.

Формат входных данных

В первой строке входных данных находятся три целых числа m , n и k ($1 \leq m, n \leq 200\,000$; $1 \leq m \cdot n \leq 200\,000$; $2 \leq k \leq 200\,000$) — количество групп школьников, которые будут заниматься в кабинете, количество парт, которые необходимо купить, и количество типов парт соответственно.

В каждой из следующих k строк находятся по два целых числа L_i и R_i ($1 \leq L_i \leq R_i \leq 10^9$), характеризующие диапазон роста школьников, для которых подходят парты типа i .

В каждой из следующих m строк находится описание группы. Каждое описание состоит из $2n$ целых чисел h_1, h_2, \dots, h_{2n} , задающих значение роста каждого из $2n$ школьников группы ($1 \leq h_i \leq 10^9$).

Формат выходных данных

В единственной строке выходных данных выведите P — минимальную величину суммарного неудобства, которую можно достичь при оптимальной покупке парт.

Система оценки

Подз.	Баллы	Ограничения				Необх. подзадачи	Результаты во время тура
		m	n	k	Доп. огр.		
1	10	$m \leq 100$	$n = 1$	$k \leq 50$	—	—	потестовые
2	10	$m = 1$	$n \leq 1000$	$k \leq 50$	—	—	потестовые
3	10	$m \leq 50$	$n \leq 5$	$k \leq 3$	—	—	потестовые
4	10	$m \leq 100$	$n \leq 1000$	$k = 2$	—	—	потестовые
5	10	$m \leq 100$	$n \leq 1000$	$k \leq 3$	—	3, 4	потестовые
6	10	$m \leq 100$	$n \leq 1000$	$k \leq 50$	$L_i = R_i$	—	первая ошибка
7	10	$m \leq 100$	$n \leq 1000$	$k \leq 50$	—	У, 1–6	первая ошибка
8	8				$L_i = R_i$	6	первая ошибка
9	8	$m \leq 100$			—	У, 1–7	первая ошибка
10	10		$n \leq 100$		—	У, 1, 3	первая ошибка
11	4				—	У, 1–10	первая ошибка

Примеры

стандартный ввод	стандартный вывод
1 2 2 5 25 50 90 60 5 10 40	10
2 3 3 200 400 300 500 100 600 300 330 440 40 30 300 150 250 350 450 550 300	130
1 3 4 10 100 200 200 10 100 300 1000 5 10 20 15 200 90	105

Пояснение к примеру

В первом примере есть только одна группа школьников, занимающаяся в классе. Следует купить по одной парте каждого вида и посадить школьников с ростами 5 и 10 за парту первого типа, а школьников с ростами 40 и 60 за парту второго типа. В таком случае неудобно сидеть будет только школьнику с ростом 40 и соответствующая величина неудобства будет равна 10.

Задача 7. Робогольф

Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

На всемирной олимпиаде роботов проводятся соревнования по робоголфу. Поле, на котором происходит игра, имеет вид прямоугольника, состоящего из $n \times m$ единичных квадратов. Строки поля пронумерованы числами от 1 до n , а столбцы — от 1 до m . Таким образом, каждый квадрат характеризуется двумя положительными целыми числами r и c — номерами строки и столбца, на пересечении которых он находится.

Два робота по очереди перемещают специальную фишку по полю, в некоторых квадратах которого находятся ловушки. Если фишка находится в квадрате (r, c) , то робот, выполняющий очередной ход, может переместить её в квадрат $(r + 1, c)$ или в квадрат $(r, c + 1)$. Если соответствующего квадрата не оказалось, поскольку фишка находится в последней строке или в последнем столбце поля, робот может переместить её за границу поля. Игра заканчивается в том случае, когда фишка оказывается либо за границей поля, либо в квадрате с ловушкой.

Каждой ловушке соответствует некоторое целое число v_i — её цена. Результат игры равен значению цены ловушки, в которой закончилась игра, или равен 0, если фишка оказалась за границей поля. Робот, который делает ход первым, стремится минимизировать результат игры, а робот, который делает ход вторым — максимизировать.

Пусть фишка вначале находится в квадрате (r, c) . Гарантированный результат игры $g(r, c)$ для этого квадрата равен минимальному возможному результату игры, которого может добиться делающий первый ход робот вне зависимости от действий соперника. Поскольку стартовый квадрат до начала матча неизвестен, разработчики роботов хотят вычислить сумму значений $g(r, c)$ для всех квадратов поля.

Требуется написать программу, которая по заданным расположениям и ценам ловушек определит сумму гарантированных результатов игры по всем квадратам поля.

Формат входных данных

Первая строка входных данных содержит три целых числа n , m и k ($1 \leq n, m \leq 10^9$; $1 \leq k \leq \min(n \cdot m, 10^5)$) — количество строк, количество столбцов и количество ловушек, соответственно.

Следующие k строк содержат по три целых числа r_i , c_i и v_i ($1 \leq r_i \leq n$, $1 \leq c_i \leq m$, $-10^9 \leq v_i \leq 10^9$) — номер строки, номер столбца и цену i -й ловушки соответственно. Никакие две ловушки не находятся в одном и том же квадрате.

Формат выходных данных

Выведите одно целое число — остаток от деления суммы гарантированных результатов игры по всем квадратам поля на число 998 244 353.

Остаток от деления a на b , где a может быть отрицательным, равен числу $r = a \bmod b$, лежащему в диапазоне от 0 до $b-1$, такому что $a = qb+r$, где q — целое. Например, $13 \bmod 4 = 1$, $-13 \bmod 4 = 3$, $12 \bmod 4 = 0$, $-12 \bmod 4 = 0$.

Примеры

стандартный ввод	стандартный вывод
3 3 3 2 3 -2 3 1 3 1 2 1	998244352
2 4 3 1 2 2 2 4 -3 2 1 1	998244348

Пояснение к примеру

В первом примере гарантированные результаты игр для всех квадратов выглядят так (квадраты с ловушками выделены):

	1	2	3
1	1	1	-2
2	0	-2	-2
3	3	0	0

Сумма результатов равна: $1 + 1 - 2 + 0 - 2 - 2 + 3 + 0 + 0 = -1$. Ответ равен $(-1) \bmod 998\,244\,353 = -1 + 998\,244\,353 = 998\,144\,352$.

В втором примере гарантированные результаты игр для всех квадратов выглядят так (квадраты с ловушками выделены):

	1	2	3	4
1	1	2	0	-3
2	1	0	-3	-3

Сумма результатов равна: $1 + 2 + 0 - 3 + 1 + 0 - 3 - 3 = -5$. Ответ равен $(-5) \bmod 998\,244\,353 = 998\,244\,348$.

Система оценки

Подз.	Баллы	Ограничения			Необх. подзадачи	Результаты во время тура
		n, m	k	дополнительно		
1	20	$n, m \leq 1\,000$			У	первая ошибка
2	14	$n \leq 5, m \leq 10^9$			У	первая ошибка
3	14	$n, m \leq 100\,000$	$k = n + m - 1$	для любого i , $r_i = n$ или $c_i = m$	—	первая ошибка
4	10			для любого i , $r_i \geq n - 1\,000$ и $c_i \geq m - 1\,000$	У, 1	первая ошибка
5	6	$n, m \leq 100\,000$		для любого i , $v_i = 1$	—	первая ошибка
6	6			для любого i , $v_i = 1$	5	первая ошибка
7	10	$n, m \leq 100\,000$			У, 1, 3, 5	первая ошибка
8	10		$k \leq 1\,000$		У	первая ошибка
9	10				У, 1–8	первая ошибка

Задача 8. Поиск идеи

Ограничение по времени: 5 секунд
Ограничение по памяти: 1024 мегабайта

3019 год. При раскопках древнего города Иннополис археологи обнаружили артефакт — жёсткий диск, на котором находится файл, предположительно содержащий тексты всех задач всероссийских олимпиад по информатике.

После исследования файла было выяснено, что информация в нём закодирована таким образом, что записанный в файле текст представляет собой строку t из букв английского алфавита. Текст с задачами оказался довольно длинным и содержал много повторений, поэтому файл хранился на диске в сжатом виде. Для его распаковки используется следующий алгоритм.

В процессе распаковки формируется строка t из строчных букв английского алфавита. Исходно строка пуста. Сжатый файл состоит из n блоков, которые должны быть обработаны в порядке следования. Каждый блок имеет один из двух типов.

- Блок «1 w », где w — строка. При обработке такого блока в конец строки t дописывается строка w .
- Блок «2 $pos\ len$ », где pos и len — положительные целые числа. Пусть символы строки t пронумерованы с 1. При обработке такого блока в конец строки t по очереди приписываются len подряд идущих символов строки t , начиная с позиции pos . При этом, если значение len достаточно велико, некоторые только что выписанные символы могут быть снова использованы при обработке того же блока.

Ученые решили выяснить, сколько раз некоторая идея встречалась в олимпиадах. Для этого они сформировали строку p из строчных букв английского алфавита и хотят найти количество вхождений строки p как подстроки в полученную после распаковки файла строку t .

Строка p длины m входит в строку t как подстрока с позиции i , если m следующих подряд символов строки t , начиная с i -го, представляют собой строку p . Например, строка «aba» входит как подстрока в строку «ababaaba» три раза: с позиций 1, 3 и 6.

Требуется написать программу, которая определяет количество вхождений заданной строки p в полученную после распаковки файла строку t .

Формат входных данных

В первой строке находятся натуральные числа m и n — длина строки p и количество блоков в сжатом тексте ($1 \leq m \leq 2 \cdot 10^5$, $1 \leq n \leq 10^4$).

Во второй строке входных данных задана непустая строка p , состоящая из строчных букв английского алфавита.

В следующих n строках находятся описания блоков в описанном в условии формате. Для блоков первого типа приписываемая строка w непуста, сумма длин всех строк w в блоках первого типа не превышает $2 \cdot 10^5$. Для блоков второго типа в строке t к моменту обработки этого блока находится не менее pos символов. Длина распакованного текста не превышает 10^{15} символов.

Формат выходных данных

Выведите одно целое число — количество вхождений строки p в текст.

Пример

стандартный ввод	стандартный вывод
3 4 aba 1 ab 2 1 3 2 3 3 2 1 8	6

Пояснение к примеру

При распаковке файла в примере последовательно получаются следующие строки:

«» → «ab» → «ababa» → «ababaaba» → «ababaabaababaaba».

Строка «aba» входит как подстрока в результирующую строку «ababaabaababaaba» 6 раз.

Система оценки

Обозначим длину текста t после обработки i блоков как L_i .

Обозначим тип i -го блока как $type_i$. Если $type_i = 2$, то обозначим параметры этого блока как pos_i и len_i .

Подз.	Баллы	Ограничения			Необх. подзадачи	Результаты во время тура
		m, n	L_n	дополнительно		
1	6	$m \leq 2000$ $n = 1$	$L_n \leq 1000$		—	первая ошибка
2	10	$m \leq 10^5$ $n \leq 2000$	$L_n \leq 10^6$		У, 1	первая ошибка
3	10	$m \leq 2000$ $n \leq 2000$	$L_n \leq 10^{10}$	кроме первого блока $type_i = 2, pos_i = 1$ len_i делится на L_1	1	первая ошибка
4	10	$m \leq 2000$ $n \leq 2000$	$L_n \leq 10^{10}$	$pos_i = L_{i-1}$	1	первая ошибка
5	10	$m \leq 20$ $n \leq 2000$	$L_n \leq 10^{10}$	$pos_i = 1, len_i \leq 10^7$	—	первая ошибка
6	4	$m \leq 2000$ $n \leq 2000$	$L_n \leq 10^{10}$	$pos_i = 1, len_i \leq 10^7$	1, 5	первая ошибка
7	10	$m \leq 20$ $n \leq 20$	$L_n \leq 10^{10}$	p состоит из букв «а» $pos_i + len_i - 1 \leq L_{i-1}$	—	первая ошибка
8	6	$m \leq 20$ $n \leq 20$	$L_n \leq 10^{10}$	$pos_i + len_i - 1 \leq L_{i-1}$	7	первая ошибка
9	2	$m = 1$ $n \leq 2000$	$L_n \leq 10^{10}$	p состоит из букв «а»	—	первая ошибка
10	4	$m \leq 20$ $n \leq 2000$	$L_n \leq 10^{10}$	p состоит из букв «а»	7–9	первая ошибка
11	5	$m \leq 20$ $n \leq 2000$	$L_n \leq 10^{10}$		У, 5, 7–10	первая ошибка
12	14	$m \leq 10^5$ $n \leq 2000$	$L_n \leq 10^{10}$		У, 1–11	первая ошибка
13	9	$m \leq 2 \cdot 10^5$ $n \leq 10000$	$L_n \leq 10^{15}$		У, 1–12	первая ошибка