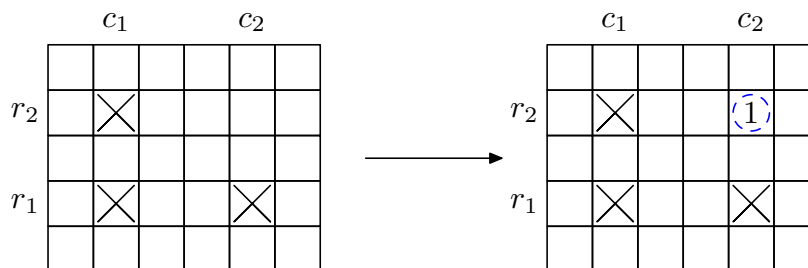


Задача А. Химическая таблица

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Учёные Иннополиса продолжили исследование периодической таблицы. Существуют $n \cdot m$ известных элементов, и они представлены в периодической таблице — прямоугольнике, состоящем из n строк и m столбцов. Каждый элемент может быть описан своими координатами в таблице (r, c) ($1 \leq r \leq n, 1 \leq c \leq m$).

Недавно учёные открыли, что для каждого четырёх различных элементов в этой таблице, которые образуют прямоугольник со сторонами, параллельными сторонам таблицы, если они имеют экземпляры трёх из четырёх элементов, то с помощью ядерного синтеза они могут произвести четвёртый элемент. Так, если имеются элементы с позиций (r_1, c_1) , (r_1, c_2) , (r_2, c_1) , где $r_1 \neq r_2$ и $c_1 \neq c_2$, то можно произвести элемент (r_2, c_2) .



Использованные экземпляры элементов не выбрасываются и могут быть использованы в дальнейшем для создания других элементов. Созданные элементы также могут в этом участвовать.

Учёные Иннополиса уже имеют образцы q элементов. Они хотят получить образцы всех $n \cdot m$ элементов таблицы. Чтобы добиться этого, они могут купить некоторые образцы в других лабораториях, а потом произвести остальные в некотором порядке. Помогите им определить, какое минимальное число элементов им надо купить.

Формат входных данных

Первая строка входных данных содержит три целых числа n, m, q ($1 \leq n, m \leq 200\,000$; $0 \leq q \leq \min(n \cdot m, 200\,000)$) — размеры таблицы элементов и количество элементов, которые учёные уже имеют.

Следующие q строк содержат по два целых числа каждая: r_i, c_i ($1 \leq r_i \leq n, 1 \leq c_i \leq m$), которые описывают расположение уже имеющихся элементов в таблице.

Формат выходных данных

Выведите минимальное количество элементов, которые надо купить.

Примеры

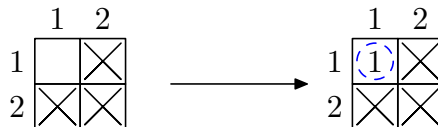
стандартный ввод	стандартный вывод
2 2 3 1 2 2 2 2 1	0
1 5 3 1 3 1 1 1 5	2
4 3 6 1 2 1 3 2 2 2 3 3 1 3 3	1

Замечание

Каждый пример имеет иллюстрацию возможного решения. На левой части картинки крестиками показаны уже имеющиеся элементы. Правая часть картинки показывает, как оставшиеся элементы могут быть получены. Красные кружки обозначают купленные элементы, а числа в синих кружках обозначают возможный порядок, в котором эти элементы могут быть получены

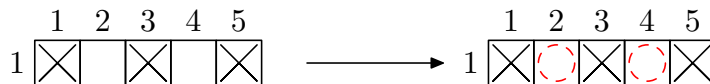
Тест 1

Мы можем получить недостающий элемент ядерным синтезом, поэтому не требуется ничего покупать.



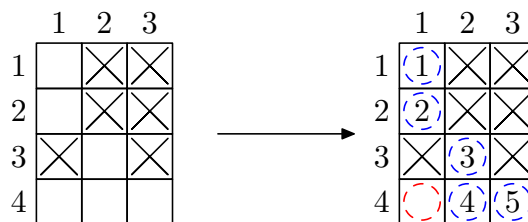
Тест 2

Так как в таблице всего один ряд, ядерный синтез не может производиться, поэтому мы вынуждены купить недостающие элементы.



Тест 3

Существует несколько возможных решений. Одно из них описано ниже.



Заметим, что непосредственно после покупки обозначенного красным элемента мы все еще не можем произвести элемент, обозначенный числом 4. Мы сможем это сделать только после получения элемента, обозначенного 1.

Задача В. Недостающие числа

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

Chouti работает над странной математической задачей.

Была некоторая последовательность из n положительных целых чисел x_1, x_2, \dots, x_n , где число n чётное. Последовательность была очень особенной, а именно для каждого целого t от 1 до n , $x_1 + x_2 + \dots + x_t$ является квадратом некоторого целого числа (то есть полным квадратом).

Неведомым образом, числа с нечётными индексами пропали, то есть известны только числа с чётными индексами, другими словами $x_2, x_4, x_6, \dots, x_n$. Задача состоит в том, чтобы восстановить изначальную последовательность. Он снова просит у вас помощи с решением этой задачи.

Автор задачи может допускать ошибки, поэтому подходящая последовательность может и не существовать. Если же существует несколько подходящих последовательностей, выведите любую.

Формат входных данных

В первой строке записано одно целое чётное число n ($2 \leq n \leq 10^5$).

Во второй строке записаны $\frac{n}{2}$ положительных целых чисел x_2, x_4, \dots, x_n ($1 \leq x_i \leq 2 \cdot 10^5$).

Формат выходных данных

Если не существует ни одной подходящей последовательности, выведите «No».

Иначе, выведите «Yes» и n положительных целых чисел x_1, x_2, \dots, x_n ($1 \leq x_i \leq 10^{13}$) в следующей строке, при этом x_2, x_4, \dots, x_n должны совпадать с числами во входных данных. Если существует несколько возможных ответов, выведите любой.

Заметьте, что ограничение на x_i больше, чем во входных данных. Можно доказать, что если ответ существует, то существует и последовательность, удовлетворяющая $1 \leq x_i \leq 10^{13}$.

Примеры

стандартный ввод	стандартный вывод
6 5 11 44	Yes 4 5 16 11 64 44
2 9900	Yes 100 9900
6 314 1592 6535	No

Замечание

В первом примере

- $x_1 = 4$
- $x_1 + x_2 = 9$
- $x_1 + x_2 + x_3 = 25$
- $x_1 + x_2 + x_3 + x_4 = 36$
- $x_1 + x_2 + x_3 + x_4 + x_5 = 100$
- $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 144$

Все эти числа являются полными квадратами.

Во втором примере, $x_1 = 100$, $x_1 + x_2 = 10000$. Эти числа являются полными квадратами. Существуют и другие возможные ответы. Например, $x_1 = 22500$ является другим возможным ответом.

В третьем примере можно показать, что подходящей последовательности не существует.

Задача С. Новая прогулка Амальтеи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Когда она была ещё княжной в небольшом шахматном замке, Амальтея каждый день гуляла по золотому церемониальному залу. Пол церемониального зала представлял собой клетчатую плоскость, разделенную на единичные квадраты. Маршрут Амальтеи был одинаковый каждый день: она вставала на определённую клетку, делала определённые переходы с текущей клетки на соседнюю по стороне и в конце возвращалась на клетку, с которой начала. Поскольку в детстве Амальтее ничего не запрещали, она могла посещать одну и ту же клетку несколько раз.

За много лет таких прогулок золото на клетках, посещаемых Амальтеей, стёрлось и стало выглядеть блёклым. Этот узор Амальтея запомнила на всю жизнь.

Теперь, став графиней в большом шахматном дворце, она решила в память о детстве выложить серебряными плитками в золотом церемониальном зале тот же самый узор.

Дворцовый скульптор, чтобы порадовать госпожу, выложил серебром тот самый узор, но увеличенный в два раза: каждой блёклой клетке из воспоминаний детства теперь соответствует квадрат 2×2 из серебряных клеток. Для каждой клетки (x, y) из этого набора он выложил серебром клетки $(2x, 2y)$, $(2x, 2y + 1)$, $(2x + 1, 2y)$ и $(2x + 1, 2y + 1)$.

Скульптора уже казнили, но теперь Амальтея хочет научиться обходить новый серебряный набор клеток. Она снова будет ходить из клетки только на соседнюю по стороне, снова должна в конце вернуться на клетку, из которой начала, но теперь, поскольку дворцовый этикет требует соблюдения строгих правил, она не имеет права посещать одну и ту же клетку дважды, за исключением последнего шага на начальную клетку.

Чтобы избежать казни дворцового математика, постройте корректный маршрут для Амальтеи.

Формат входных данных

В первой строке задано целое число n — количество блёклых клеток в узоре из детства ($1 \leq n \leq 30\,000$).

Каждая из последующих n строк содержит два целых числа x_i, y_i — координаты блёклой клетки ($0 \leq x_i, y_i \leq 1000$).

Гарантируется, что никакая клетка не повторяется, и что область из этих клеток является связанной по стороне.

Формат выходных данных

Выведите $4n$ строк, в каждой из них выведите координаты очередной клетки маршрута Амальтеи во дворце. Каждые две соседние клетки, а также первая и последняя в выведенном маршруте должны быть соседними по стороне. В маршрут должны войти все серебряные клетки.

Пример

стандартный ввод	стандартный вывод
3	0 0
0 0	0 1
0 1	0 2
1 0	0 3
	1 3
	1 2
	1 1
	2 1
	3 1
	3 0
	2 0
	1 0

Задача D. Игра со стеком

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

У Вани есть натуральное число n . Он придумал следующую игру. У него есть стек, в котором изначально лежит единственное число 1. Он может выполнять следующую операцию: пусть сейчас в стеке лежит s целых чисел a_1, a_2, \dots, a_s , в порядке от самого первого элемента стека к самому последнему. Ваня может выбрать два целых числа $1 \leq l \leq r \leq s$ и положить число $a_l + a_{l+1} + \dots + a_r$ в конец стека.

Ваня хочет сделать последнее число в стеке равным n . Определите минимальное количество операций, которое для этого требуется и постройте любой возможный пример с этим количеством операций.

Формат входных данных

В первой строке находится единственное целое число t ($1 \leq t \leq 1000$) — количество тестовых случаев.

В следующих t строках находится по одному целому числу n ($1 \leq n \leq 10^{18}$) — число Вани.

Формат выходных данных

Выведите ответы на тестовые случаи в порядке их следования во входных данных. Для тестового случая выведите ответ в следующем формате:

В первой строке выведите единственное целое число q ($0 \leq q \leq 1000$) — минимальное количество операций со стекком, которое нужно сделать, чтобы последнее число стало равно n . Гарантируется, что за ≤ 1000 операций возможно сделать последнее число равным n . В следующих строках выведите по два целых числа — описания операций. В i -й строке выведите два целых числа l_i, r_i ($1 \leq l_i \leq r_i \leq i$). Во время i -й операции в конец стека кладется число $a_{l_i} + a_{l_i+1} + \dots + a_{r_i}$. После выполнения предоставленных операций последнее число стека должно быть равно n .

Все тестовые случаи независимы.

Пример

стандартный ввод	стандартный вывод
3	2
2	1 1
3	1 2
7	3
	1 1
	2 2
	1 3
	4
	1 1
	1 2
	2 3
	1 4

Задача E. Минимальная уникальная подстрока

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

Пусть у вас есть строка s из символов “0” и “1”. Будем называть строку t подстрокой строки s , если существует $1 \leq l \leq |s| - |t| + 1$, такое что $t = s_l s_{l+1} \dots s_{l+|t|-1}$. Будем называть подстроку t строки s уникальной, если существует единственное такое l .

Например, пусть $s = \text{“1010111”}$. Тогда $t = \text{“010”}$ является уникальной подстрокой s , так как существует единственное подходящее $l = 2$. Заметим, что $t = \text{“10”}$ не является уникальной подстрокой s , так как подходят $l = 1$ и $l = 3$. А, например, $t = \text{“00”}$ вообще не является подстрокой строки s , так как не существует подходящих l .

Сегодня Вася на уроке информатики решал такую задачу: дана строка из символов “0” и “1”, надо найти длину её кратчайшей уникальной подстроки. Написав решение к этой задаче, он решил его протестировать. Он просит помощи у вас.

Вам даны 2 таких целых положительных числа n и k , что $(n \bmod 2) = (k \bmod 2)$, где $(x \bmod 2)$ — это операция взятия остатка числа x при делении на 2. Найдите любую строку s состоящую из n символов “0” и “1”, такую что наименьшая длина её уникальной подстроки равна k .

Формат входных данных

В первой строке даны два целых числа n и k , разделённые пробелом ($1 \leq k \leq n \leq 100\,000$, $(k \bmod 2) = (n \bmod 2)$).

Формат выходных данных

Выведите строку s длины n , состоящую из символов “0” и “1”. Минимальная длина уникальной подстроки s должна равняться k . Среди таких строк разрешается вывести **любую**. Гарантируется, что хотя бы одна подходящая строка существует.

Примеры

стандартный ввод	стандартный вывод
4 4	1111
5 3	01010
7 3	1011011

Замечание

В первом тесте легко видеть, что единственной уникальной подстрокой строки $s = \text{“1111”}$ является вся строка s , длина которой 4.

Во втором тесте у строки $s = \text{“01010”}$ минимальной по длине уникальной подстрокой является строка $t = \text{“101”}$ длина которой 3.

Во третьем тесте у строки $s = \text{“1011011”}$ минимальной по длине уникальной подстрокой является строка $t = \text{“110”}$ длина которой 3.

Задача F. Головоломка из фишек

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

Егор придумал новую головоломку из фишек, в которую предлагает сыграть вам.

Головоломка имеет вид таблицы из n строк и m столбцов, в каждой клетке которой могут располагаться несколько черных и белых фишек, положенных в ряд. Таким образом, состояние в клетке можно описать строкой, состоящей из символов «0» (белая фишка) и «1» (черная фишка), возможно пустой, а вся головоломка — это таблица, в каждой клетке которой стоит строка из нулей и единиц. Задача состоит в том, чтобы из одного состояния таблицы получить другое.

Для этого можно использовать следующую операцию:

- выбрать две различные клетки (x_1, y_1) и (x_2, y_2) , при этом клетки должны находиться в одной строке или одном столбце таблицы, и строка в клетке (x_1, y_1) должна быть непустой;
- за операцию можно переместить последний символ строки в клетке (x_1, y_1) в начало строки в клетке (x_2, y_2) .

Для вас Егор загадал 2 состояния таблицы — начальное и конечное. Гарантируется, что количества нулей и единиц в таблицах совпадают. Ваша задача — с помощью нескольких операций получить из начального состояния конечное. Конечно, Егор не хочет, чтобы количество операций было очень большим. Обозначим за s количество символов в каждой из таблиц (в таблицах поровну символов). Тогда вы должны использовать не более $4 \cdot s$ операций.

Формат входных данных

В первой строке записаны два целых числа n и m ($2 \leq n, m \leq 300$) — количества строк и столбцов у таблиц в головоломке, соответственно.

В следующих n строках находится описание начального состояния таблицы в следующем формате: в каждой строке находится m непустых строк, состоящих из нулей и единиц. В i -й из этих строк, j -я строка описывает строку, записанную в клетке (i, j) . Строки нумеруются от 1 до n , столбцы нумеруются от 1 до m .

В следующих n строках находится описание конечного состояния таблицы в аналогичном формате.

Обозначим суммарную длину строк в начальном состоянии за s . Гарантируется, что $s \leq 100\,000$. Также гарантируется, что количества нулей и единиц совпадают в начальном и конечном состояниях.

Формат выходных данных

В первой строке выведите одно целое число q — количество использованных операций. Необходимо найти решение, для которого $0 \leq q \leq 4 \cdot s$.

В следующих q строках выведите по 4 целых числа x_1, y_1, x_2, y_2 . i -я из этих строк должна описывать i -ю операцию. Необходимо, чтобы было выполнено $1 \leq x_1, x_2 \leq n$, $1 \leq y_1, y_2 \leq m$, $(x_1, y_1) \neq (x_2, y_2)$, $x_1 = x_2$ или $y_1 = y_2$. При этом строка в клетке (x_1, y_1) должна быть непустой. Данная последовательность операций должна переводить таблицу из начального состояния в конечное.

Можно показать, что ответ существует. Если есть несколько возможных решений, выведите любое.

Примеры

стандартный ввод	стандартный вывод
2 2	4
00 10	2 1 1 1
01 11	1 1 1 2
10 01	1 2 2 2
10 01	2 2 2 1
2 3	4
0 0 0	2 2 1 2
011 1 0	1 2 2 2
0 0 1	1 2 1 3
011 0 0	1 3 1 2

Замечание

Рассмотрим первый пример.

- Текущее состояние таблицы:

```
00 10
01 11
```

Первая операция. В клетке (2,1) записана строка 01. Применяя операцию к двум клеткам (2,1) и (1,1), мы переносим 1 из конца строки 01 в начало строки 00, получая строку 100.

- Текущее состояние таблицы:

```
100 10
0 11
```

Вторая операция. В клетке (1,1) записана строка 100. Применяя операцию к двум клеткам (1,1) и (1,2), мы переносим 0 из конца строки 100 в начало строки 10, получая строку 010.

- Текущее состояние таблицы:

```
10 010
0 11
```

Третья операция. В клетке (1,2) записана строка 010. Применяя операцию к двум клеткам (1,2) и (2,2), мы переносим 0 из конца строки 010 в начало строки 11, получая строку 011.

- Текущее состояние таблицы:

```
10 01
0 011
```

Четвертая операция. В клетке (2,2) записана строка 011. Применяя операцию к двум клеткам (2,2) и (2,1), мы переносим 1 из конца строки 011 в начало строки 0, получая строку 10.

- Текущее состояние таблицы:

```
10 01
10 01
```

Можно видеть, что мы получили требуемое состояние таблицы.

Задача G. 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 контекст 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 ($1 \leq l_i \leq r_i \leq n$), ($1 \leq x_i \leq n$) и означает, что к элементам с номерами $l_i, l_i + 1, \dots, r_i$ 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 ($1 \leq n, q \leq 10^4$) — 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 ($1 \leq l_i \leq r_i \leq n$, $1 \leq x_i \leq n$), 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).