

## Problem A. Арифметическая производная - 1

Input file:            стандартный ввод  
Output file:         стандартный вывод  
Time limit:          2 секунды  
Memory limit:       256 мегабайт

И какой же математик не любит производную? И что за неведомая сила заключена в ней? Её ли не любить, когда в ней слышится что-то восторженно-чуждое? И не хитрая, кажется, операция — подсчитать производную, а какое истинное наслаждение испытываешь, вычисляя её.

Дифференцированию можно обучить любого человека, нужно лишь объяснить ему определение производной. Начнём с дифференцирования... натуральных чисел. Значение производной для каждого простого числа  $p$  будем считать равным 1, а для составного числа  $n$  вычислим это значение по правилу Лейбница:  $n' = (a \cdot b)' = a' \cdot b + a \cdot b'$ . Например,  $6' = (2 \cdot 3)' = 2' \cdot 3 + 2 \cdot 3' = 5$ .

Вам необходимо найти все натуральные числа, не превосходящие  $r$ , для которых производная  $n'$  больше исходного числа  $n$  ровно в  $k$  раз.

### Input

Входные данные содержат два целых числа  $k$  и  $r$  ( $1 \leq k \leq 30, 1 \leq r \leq 2 \cdot 10^{18}$ ).

### Output

Если требуемых чисел нет, выведите число 0. Иначе в первой строке выведите одно целое  $m$  — количество целых положительных чисел  $n$ , не превосходящих  $r$ , для которых  $n' = k \cdot n$ . Во второй строке запишите эти  $m$  чисел в порядке возрастания.

### Examples

стандартный ввод	стандартный вывод
1 100	2 4 27
1 2	0

## Problem B. Белый треугольник

Input file: *standard input*  
Output file: *standard input*  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

Под огромной плоской поляной находится месторождение редкого ископаемого белого цвета, очень напоминающего снег. О месторождении точно известно, что оно имеет форму треугольника.

В точках  $A$  и  $B$  были сооружены геологические скважины. Погруженные в скважины зонды позволили измерить точное расстояние от скважины до прямых, на которых лежат стороны треугольника. Теперь необходимо обработать полученный массив данных и выяснить координаты вершин подземного белого треугольника.

Скважины могут располагаться как внутри, так и вне треугольника, но не могут лежать на его сторонах. Расстояния от скважин внутри треугольника считаются положительными, а для скважин вне треугольника — отрицательными.

### Input

В двух первых строках записаны координаты скважины  $A$  и расстояния от неё до прямых, на которых лежат стороны  $XY$ ,  $YZ$  и  $ZX$  треугольника  $XYZ$ . В следующих двух строках — координаты скважины  $B$  и расстояния до тех же прямых  $XY$ ,  $YZ$  и  $ZX$ . Координаты  $A$  и  $B$  — целые числа, не превосходящие по абсолютной величине  $10^4$ . Все расстояния — целые ненулевые числа, не превосходящие по абсолютной величине  $10^4$ . Гарантируется, что треугольник  $XYZ$  существует и координаты его вершин не превосходят по абсолютной величине  $10^9$ .

### Output

В трёх строках запишите через пробел координаты вершин  $X$ ,  $Y$  и  $Z$  треугольника. Верным считается любой ответ, где для точек  $A$  и  $B$  расстояния до сторон найденного треугольника отличаются от требуемых не более, чем на  $10^{-5}$ .

### Examples

standard input	standard input
1 5	0.00000 0.00000
1 5 5	0.00000 12.0000
5 2	16.0000 0.00000
5 5 2	
5 5	20.0000 0.00000
5 5 5	0.00000 0.00000
5 -5	0.00000 15.0000
-5 -5 -13	

## Problem C. Мы передумали

Input file: *standard input*  
Output file: *standard input*  
Time limit: 5 секунд  
Memory limit: 256 мебибайт

В архитектурское бюро пришло задание на проектирование улицы. Всего на улице имеется  $n$  участков под застройку — все по одну сторону дороги. На каждом участке будет построен дом.

Бюро должно предоставить проект, содержащий планируемое количество этажей для каждого дома. Однако требования на суммарное количество этажей а также на минимальную и максимальную высоту дома на некоторых участках постоянно меняются.

Требуется написать программу, которая будет находить количество различных проектов, удовлетворяющих текущим ограничениям. Два проекта считаются различными, если существует хотя бы один участок, для которого в соответствующих проектах планируемое количество этажей различно.

Необходимо обработать  $m$  запросов вида:

- поменять ограничения на количество этажей домов на участках с  $l$  по  $r$
- изменить требование на суммарное количество этажей

### Input

В первой строке три целых числа  $n$ ,  $A$  и  $m$  ( $1 \leq n \leq 10^9, 1 \leq A \leq 10^3, 1 \leq m \leq 10^5$ ) — количество участков на улице, суммарное количество этажей и количество запросов соответственно.

Следующие  $m$  строк содержат запросы. Каждая строка начинается числом  $t$ , означающим тип операции. Если  $t$  равно 1, то после него через пробел записано 4 целых числа  $l$ ,  $r$ ,  $C$ ,  $D$  ( $1 \leq l \leq r \leq n, 0 \leq C \leq D \leq 10^3$ ) — ограничение на высоту, означающее что на участках с  $l$  по  $r$  допустимо строить дома не ниже  $C$  этажей, но не выше  $D$  этажей.

Если  $t$  равно 2, то после него через пробел записано 1 целое число  $A$  ( $0 \leq A \leq 10^3$ ) — новое требование на суммарное количество этажей.

Количество запросов первого типа не превышает 30.

### Output

После каждого запроса выведите количество различных проектов по модулю 1000210433.

## Examples

standard input	standard input
2 2 3 1 2 2 2 3 2 3 1 2 2 0 3	1 2 4
3 4 7 1 1 1 2 10 2 0 2 5 2 7 1 1 2 3 3 2 5 2 6	6 0 10 21 1 0 1
1000000 9 6 2 0 2 1 2 1000 1 1 100 10 10 2 1 2 1000	1 1000000 860010478 1 0 1

## Problem D. Иван-царевич и Армия Клонов

Input file: *standard input*  
Output file: *standard input*  
Time limit: 1 секунда  
Memory limit: 256 мебибайт

Иван-царевич, отправившись спасать свою невесту Василису Прекрасную из лап Кощея Бессмертного, был телепортирован Бабой-Ягой прямо ко входу в сокровищницу Кощея. Вредная старуха обещает рассказать, где находится смерть Кощея, только если Иван-царевич сможет выйти из сокровищницы живым, захватив оттуда как можно больше золотых монет. К счастью, у Ивана Царевича в руках оказался подробный план сокровищницы.

Сокровищница состоит из  $n$  залов. У Ивана-царевича есть волшебное яблочко на блюдечке, которое может помочь ему пройти по секретным ходам от входа в сокровищницу в любой из залов, какой он пожелает. Залы соединены таким образом, что попав в сокровищницу, из 1-го зала можно перейти только во 2-й, из 2-го в третий и т.д. Назад возвращаться нельзя, поскольку это дестабилизирует волшебное поле и разрушит сокровищницу.

В каждом зале находится либо одна золотая монета, либо одноразовая шапка-невидимка, либо клон Кощея. Проходя через зал, Иван-царевич поднимает и уносит с собой все монеты и шапки-невидимки. Пройти мимо клона Кощея можно только использовав одну из шапок-невидимок.

Иван-царевич может вызвать свистом верного Серого Волка, который унесет его из любого зала на свободу. К сожалению, Волка можно уговорить спуститься в сокровищницу только один раз, поскольку он страдает клаустрофобией.

Помогите Ивану-царевичу. Подсчитайте, какое максимальное количество золотых монет он может унести из сокровищницы.

### Input

Единственная строка длиной  $n$  ( $1 \leq n \leq 10^6$ ) символов содержит описание сокровищницы. Номер символа в строке соответствует номеру зала. Символ «H» обозначает зал с шапкой-невидимкой, символ «M» — зал с золотой монетой, символ «K» — зал с клоном Кощея. Строка не содержит других символов кроме вышеперечисленных.

### Output

Единственное число — максимально возможное количество собранных монет.

### Example

standard input	standard input
МКНМКМНМНМКМННМККННМКМ	3

## Problem E. Есть билет на Ганимед

Input file: *standard input*  
Output file: *standard input*  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

На различных объектах необъятной Солнечной Системы расположены  $K^2$  космических станций. Сюда в гости направляется огромная туристическая группа из перенаселенной галактики Чи. Каждому космическому туристу выдана путевка с уникальным номером, состоящим из  $N$  разрядов в системе счисления с основанием  $K$ . Путевка дает право вычеркнуть из номера путевки  $N - 2$  разряда и прибыть на станцию с получившимся номером.

Администрация космических железных дорог и межпланетных путей очень не любит Чи-галактианцев, поэтому планирует закрыть часть станций на ремонт, но так, чтобы каждый турист по-прежнему имел возможность прибыть хотя бы на одну открытую станцию. Найдите минимальное количество станций, которое способно принять всех туристов в соответствии с правилами обслуживания путевок.

### Input

Единственная строка ввода содержит натуральные числа  $N$  и  $K$  ( $2 \leq N \leq 100$ ,  $2 \leq K \leq 100$ ).

### Output

Выведите искомое количество.

### Examples

standard input	standard input
4 10	34
12 10	10

## Problem F. Игры с кольцами

Input file: *standard input*  
Output file: *standard input*  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

Саша и Маша играют в кольца.

У Саши в руках матрица цифр размером  $N$  строк и  $M$  столбцов. Саша должен разрезать матрицу на столбцы, выбрать порядок их следования и не разворачивая склеить в одну длинную ленту. Затем ленту следует склеить в кольцо и вновь разрезать в произвольном месте таким образом, чтобы на ленте сверху вниз можно было прочесть максимально возможное число.

В руках Маши идентичная матрица, однако она действует иначе. Её задача — разрезать матрицу на строки, выбрать порядок их следования и не разворачивая склеить в одну длинную ленту. Затем Маша должна склеить ленту в кольцо и вновь разрезать в произвольном месте таким образом, чтобы на ленте слева направо можно было прочесть максимально возможное число.

Выиграет тот, чье число окажется больше. Найдите максимально возможное число, которое может получиться у Саши или Маши.

### Input

В первой строке содержатся натуральные числа  $N$  и  $M$ , не превышающие 100. В каждой из последующих  $N$  строк записаны  $M$  цифр без пробелов, описывающие матрицу. Гарантируется, что в матрице есть хотя бы одна цифра, отличная от 0.

### Output

Выведите искомое число.

### Example

standard input	standard input
2 2 28 27	8722

## Problem G. Вемлепуть

Input file: *standard input*  
Output file: *standard input*  
Time limit: 1 секунда  
Memory limit: 256 мебибайт

В плоской стране протекает Молочная Река с Кисельными Берегами. Неудивительно, что основным занятием жителей страны является добыча молока. Чтобы упростить процессы транспортировки молока, его обработки и получения продуктов, было решено построить Великий Млечный Путь.

Согласно плану Великий Млечный Путь должен быть замкнутым, проходить через каждый город ровно по одному разу и использовать для соединения городов прямые дороги. Жителям плоской страны неизвестны правила дорожного движения и технологии забивания свай в кисельные берега, поэтому Путь не должен содержать перекрёстков и пересекать Реку где-либо кроме её истока.

Исток Реки и все города обозначены на карте точками с целочисленными координатами. Сама Река протекает вдоль прямой, заданной вектором с целочисленными координатами, и где-то бесконечно далеко впадает в Молочный Океан. Жители плоской страны заботятся об экологии, поэтому никогда не строили города на Реке, включая точку истока.

Требуется спроектировать Великий Молочный Путь.

### Input

В первой строке единственное целое число  $N$ , не превышающее 10000 — количество городов. Во второй строке  $X, Y, dX, dY$  — координаты истока и вектор направления реки соответственно. В следующих  $N$  строках записаны пары координат городов  $X_i, Y_i$ . Все координаты — целые числа, не превосходящие по модулю  $10^4$ . Гарантируется, что существует пара городов, между которыми можно проложить дорогу, и пара городов, между которыми нельзя проложить дорогу из-за пересечения с Рекой.

### Output

Выведите  $N$  чисел — номера городов в порядке их следования в Пути, либо единственное число -1, если построить требуемый Путь невозможно.



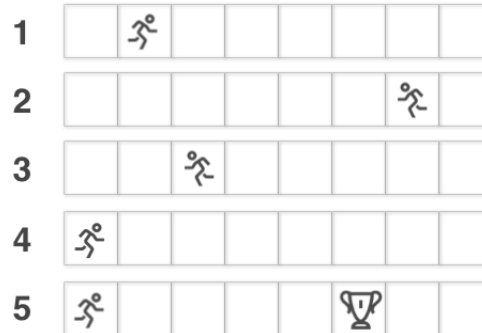
## Example

standard input	standard input
20	13 12 8 9 1 2 10 19 20 16 7 6 3 14 17
0 0 0 -1	11 18 4 5 15 13
-2 3	
2 3	
4 0	
-4 0	
-3 -2	
3 -2	
3 -4	
-6 -2	
-6 2	
6 2	
-1 1	
-5 -5	
-4 -5	
2 0	
-3 -4	
4 -5	
1 1	
-2 0	
6 -2	
5 -5	

## Problem H. Параллельно

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

Изобретён новый вид спорта — параллельная эстафета. Стадион для такой эстафеты строится следующим образом:



Каждый из спортсменов команды бежит по своей дорожке. Эстафета начинается с бегуна на первой дорожке. Его задача добежать до позиции, рядом с которой ожидает его партнер на второй дорожке и передать ему эстафету. Второй бегун аналогично передает эстафету третьему, и так далее до последнего, которому нужно только добежать до позиции финиша.

К предстоящему чемпионату мира необходимо построить стадион. Международная федерация параллельного бега предоставила технические требования к стадиону. В требованиях точно указаны стартовая позиция первого спортсмена, финиш для последнего, а также места старта спортсменов на некоторых дорожках. Все номера позиций отсчитываются от левого края стадиона начиная с единицы.

По остальным дорожкам федерация указала номера позиций, но с какой стороны стадиона (левой или правой) следует её отсчитывать, не уточнила, оставив это на усмотрение организаторов. Кроме того, если несколько таких дорожек идут подряд, то требования допускают произвольную перестановку их порядка внутри данной группы.

Поскольку наша сборная по параллельному бегу показывает лучший результат при меньшей дистанции, требуется построить такую конфигурацию стадиона, соответствующую требованиям, в которой спортсменам необходимо пробежать минимальное суммарное расстояние.

### Input

В первой строке записаны целые положительные числа  $N$ ,  $M$  — количество дорожек и их длина соответственно ( $1 \leq N, M \leq 10000$ ).

Вторая строка содержит два числа — стартовая позиция первого спортсмена и финишная позиция последнего.

Следующие  $N - 1$  строк содержат описание дорожек начиная со второй в формате «1 X» или «2 X»: «1 X» — место старта находится в позиции X, отсчитываемой от левого края стадиона; «2 X» — место старта находится в позиции X, но неизвестно от какого края стадиона вести отсчет.

### Output

Выведите минимальное расстояние, которое должны пробежать спортсмены на искомой конфигурации стадиона.

## Examples

стандартный ввод	стандартный вывод
5 8 2 6 1 7 2 3 1 1 2 1	16
5 21 4 15 2 5 2 6 1 2 2 2	19

## Problem I. Минимальный префикс

Input file: *standard input*  
Output file: *standard input*  
Time limit: 5 секунд  
Memory limit: 256 мегабайт

Даны  $N$  различных строк. Разрешается циклически сдвинуть каждую строку произвольное количество раз. Необходимо найти такой минимальный  $k$ , что существует некий набор сдвигов, при котором «префиксы» длины  $k$  у всех строк будут попарно различны.

В рамках данной задачи назовем «префиксом» длины  $k$  строки  $S$  такую функцию:

$$P(S, k) = \begin{cases} S & \text{if } |S| \leq k \\ S[1\dots k] & \text{otherwise} \end{cases}$$

### Input

В первой строке задано одно целое число  $N$  ( $2 \leq N \leq 200$ ) — количество строк. Далее следуют  $N$  строк, состоящих из строчных букв латинского алфавита. Суммарная длина строк не превышает  $2 \cdot 10^5$ .

### Output

Выведите минимальную длину «префикса».

### Examples

standard input	standard input
4 aaaaa aabac abaaa ababa	2
5 noon oon noo no onn	3

## Problem J. Полёт подождёт

Input file: *standard input*  
Output file: *standard input*  
Time limit: 2 секунды  
Memory limit: 256 мегабайт

Авиакомпания «Fly Mamai» предоставила чартер для отправки  $M$  команд программистов на соревнования. Все  $N$  программистов уже зарегистрировались на рейс, прошли предполетный досмотр и стоят в очереди на посадку. Самолет заправлен и готов к вылету. Однако посадка задерживается.

Дело в том, что согласно контракту о транспортировке программистов вознаграждение авиакомпании обратно пропорционально суммарному недовольству пассажиров. А рассчитывается это недовольство при посадке.

Перед выходом из терминала стоят два автобуса вместимостью  $K$  пассажиров каждый. Каждую секунду из очереди выходит очередной программист и садится в указанный сотрудником авиакомпании автобус. Каждый программист в очереди относится к одной из  $M$  команд. Программистов из одной команды нельзя разделять по разным автобусам. Автобус отправляется, как только в него заходит последний из пассажиров, которые должны сесть в этот автобус, и не возвращается. Назовём недовольством программиста время в секундах от начала посадки до момента отправления его автобуса.

Необходимо найти такое распределение программистов по двум автобусам, которое минимизирует их суммарное недовольство и, как следствие, максимизирует гонорар авиакомпании.

### Input

В первой строке ввода три натуральных числа  $N$ ,  $M$ ,  $K$  ( $M \leq 2000$ ,  $M \leq N \leq 10^5$ ,  $K \leq 10^5$ ). Во второй строке  $N$  чисел  $A_i$ , разделенных пробелом — номера команд, которым принадлежат программисты в порядке их следования в очереди ( $1 \leq A_i \leq M$ ).

### Output

Выведите единственное число — минимально возможное суммарное недовольство программистов или -1, если выполнить все условия невозможно.

### Examples

standard input	standard input
7 3 5 2 2 1 1 1 3 1	39
12 3 9 1 1 1 2 3 2 2 2 2 2 2	116
2 1 2 1 1	4

## Problem K. Нью-Тетрис

Input file: *standard input*  
Output file: *standard input*  
Time limit: 3 секунды  
Memory limit: 256 мебибайт

В новой версии игры «Тетрис» используется плоский клетчатый «колодец» шириной  $N$  столбцов и глубиной  $M$  строк. На дне колодца (под каждым из столбцов) записаны  $N$  чисел. В отличие от классической версии игры, в качестве падающих фигур используются только квадратики  $1 \times 1$ , на которых также записаны числа. Все числа не превосходят  $10^9$ .

Как только в верхней строке появляется очередной квадратик, он начинает опускаться со скоростью 1 строка в секунду. За эту секунду игрок должен выбрать направление перемещения квадратика: влево или вправо, либо оставить квадратик в текущем столбце. По окончании каждой секунды падающая фигура, если это возможно, сначала движется влево или вправо на одну клетку (если игрок этого пожелал), а затем на строку ниже, если это возможно. Если опуститься нельзя, то есть ниже находится дно, или в соответствующей клетке строкой ниже уже есть квадратик, то фигура завершает движение, и появляется новый квадратик.

После приземления каждой фигуры игроку демонстрируются его очки. Очки равны сумме чисел, являющихся самыми верхними для каждого из столбцов.

Игрок действует жадным образом: он не знает какой квадратик будет падать следующим, поэтому для каждого квадратика он выбирает такую позицию, чтобы максимизировать свои очки после приземления. Если таких позиций несколько, то он предпочитает ту, для достижения которой необходимо минимальное количество перемещений квадратика по горизонтали (влево или вправо). Если и таких позиций несколько, то он выбирает самую левую.

Всего появится  $K$  квадратиков и необходимо для каждого предсказать очки игрока после его падения. Гарантируется что стратегия игрока позволяет разместить все квадратики внутри колодца.

### Input

В первой строке записаны два целых числа  $N$  и  $M$  ( $1 \leq N, M \leq 10^5$ ) — ширина и глубина колодца соответственно.

Следующая строка содержит  $N$  натуральных чисел в порядке их записи на дне колодца.

В третьей строке записано количество падающих квадратиков  $K$  ( $1 \leq K \leq 10^5$ ).

В каждой из  $K$  последующих строк содержится по паре чисел, описывающих квадратик в порядке их появления:  $col_i$  и  $x_i$  ( $1 \leq col_i \leq N, 0 \leq x_i \leq 10^9$ ), где  $col_i$  — столбец, в котором появится  $i$ -тый по счету квадратик, а  $x_i$  — записанное в нем число.

### Output

Выведите  $K$  чисел — количество очков игрока после приземления каждой фигуры.

## Examples

standard input	standard input
6 4	29
5 2 3 4 5 5	29
4	29
4 7	30
6 3	
6 3	
6 5	
6 3	1
0 0 0 0 0 0	2
6	3
4 1	4
4 1	5
4 1	5
4 1	
4 1	
4 1	
4 1	

## Problem L. Пушечный пасьянс

Input file: *standard input*  
Output file: *standard input*  
Time limit: 2 секунды  
Memory limit: 256 мегабайт

На клетчатом игровом поле  $N \times M$  некоторые клетки заняты турелями. Турель — это башня с четырьмя однозарядными пушками, направленными на север, юг, запад и восток. При попадании снаряда турель активируется. Активация турели приводит к выстрелу из всех стволов и последующему саморазрушению.

Игрок может установить ровно одну турель на любую свободную клетку и активировать её. Требуется найти такую позицию для первого хода игрока, которая позволит уничтожить максимальное количество турелей.

### Input

В первой строке входного файла записаны натуральные числа  $N$ ,  $M$ , не превышающие 2000 — размерность игрового поля.

Следующие  $N$  строк содержат по  $M$  символов, описывающих содержимое клеток игрового поля: символ «+» означает, что в данной позиции расположена турель, символ «.» (точка) соответствует пустой клетке.

### Output

В первой строке выведите максимальное количество турелей, которые возможно уничтожить первым ходом, а во второй — координаты клетки (номер строки и номер столбца, разделенные пробелом), где необходимо установить турель. Если нельзя выполнить ход, уничтожающий хотя бы одну турель, вывести единственное число 0 (ноль).

### Examples

standard input	standard input
3 4 ++.. +... ..++	5 2 4
4 5 ++... ..+.. ....+ ...++	5 4 1
3 3 +++ +++ +++	0