0-1 Sequences

Problem ID: sequences
CPU Time limit: 1 second
Memory limit: 1024 MB
Difficulty: 6.5

Author: Tung Kam Chuen
Source: Hong Kong Regional Online Preliminary 2016

You are given a sequence, in the form of a string with characters '`0`', '`1`', and '`?`' only. Suppose there are k '`?`'s. Then there are 2^k ways to replace each '`?`' by a '`0`' or a '`1`', giving 2^k different `0-1` sequences (`0-1` sequences are sequences with only zeroes and ones).

For each `0-1` sequence, define its number of inversions as the minimum number of adjacent swaps required to sort the sequence in non-decreasing order. In this problem, the sequence is sorted in non-decreasing order precisely when all the zeroes occur before all the ones. For example, the sequence `11010` has 5 inversions. We can sort it by the following moves: `11010` → `11001` → `10101` → `01101` → `01011` → `00111`.

Find the sum of the number of inversions of the 2^k sequences, modulo $1\,000\,000\,007$ ($10^9 + 7$).

Input

The first and only line of input contains the input string, consisting of characters '`0`', '`1`', and '`?`' only, and the input string has between 1 to $500\,000$ characters, inclusive.

Output

Output an integer indicating the aforementioned number of inversions modulo $1\,000\,000\,007$.

Sample Input 1

```
?0?
```

Sample Output 1

```
3
```

2, 4, 6, Greaaat

Problem ID: 246greaaat
CPU Time limit: 1 second
Memory limit: 1024 MB
Difficulty: 8.0

Authors: Steven Halim and Robin Christopher Yu
Source: NUS Competitive Programming

Against her wishes, Rainbow Dash was put in charge of coach the buckball halftime cheerleading squad. While she doesn't have the slightest care for cheerleading (she'd much rather coach the team), she's determined to give it her all to show everypony in the squad how much she cares about them.

Figure 1: Illustration by Kyle Coate.

One of the most important things in cheerleading is regulating the crowd's *level of enthusiasm*. By instructing the squad to execute different cheers, she can raise or lower this level; for example, less lively cheers temper the level of enthusiasm, while other, more exciting cheers raise it.

At the beginning, the crowd's level of enthusiasm is zero. To prime the crowd for the second half of the game, she wants the level of enthusiasm after all the cheers to be *exactly* T. Note that it is possible for the crowd's level of enthusiasm to dip below zero or exceed T in between the cheers.

There are $N + 1$ cheers at Rainbow Dash's disposal. The i^{th} of these cheers changes the level of enthusiasm by S_i, and has a level of difficulty D_i—a higher difficulty requires more practice and is harder to execute correctly. The first of these cheers is always a standard cheer with $S_1 = 1$ and $D_1 = 1$; the other N non-standard cheers are possibly more complicated.

Help Rainbow Dash plan out a sequence of cheers such that the crowd's level of enthusiasm after all the cheers is exactly T. Note that the same cheer can be executed multiple times, and the order of the cheers can be rearranged as necessary; moreover, the existence of the standard cheer ensures that this goal is always attainable.

Among all such sequences, she wants the total level of difficulty to be minimized in order to reduce the amount of practice the squad will need—time is of the essence and we need to make the most use of it. If a cheer appears multiple times in the sequence, its level of difficulty must be included in the total that many times.

Input

The first line of input contains two integers, N $(0 \le N \le 500)$ and T $(1 \le T \le 200\,000)$, the number of non-standard cheers at Rainbow Dash's disposal and the required level of enthusiasm, respectively.

The next N lines of input contain the descriptions of the non-standard cheers. In particular, the i^{th} of these lines contains two integers S_{i+1} $(-T \le S_{i+1} \le T)$ and D_{i+1} $(1 \le D_{i+1} \le 10^9)$, denoting that the $(i+1)^{\text{th}}$ cheer changes the level of enthusiasm by S_{i+1} and has a level of difficulty of D_{i+1}.

Output

On the first line, output a single integer c, the number of cheers in the sequence.

On the second line, output c integers $d_1, d_2, \ldots, d_c$ $(1 \le d_i \le N + 1)$, denoting the cheers the squad should execute. In particular, the i^{th} cheer the squad should execute is the $d_i{}^{\text{th}}$ one. This sequence of cheers should satisfy the stipulated requirements, and, among all such sequences of cheers, have the minimum total level of difficulty.

If there are multiple correct answers, you can output any of them.

The input will be such that a solution exists with $c \le 200\,000$.

Sample Input 1

```
3 20
7 3
10 8
-2 1
```

Sample Output 1

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

Knightmare!

Given a large number of knights on an infinite chessboard, determine the number of squares that are threatened by k or more knights. Unfortunately for you, these are not normal knights! They can move in the following way. Each knight has two values, a and b. For any two integers, i $(1 \leq i \leq a)$ and j $(1 \leq j \leq b)$, the knight can move to any of the following squares relative to its current location: $(i, j), (-i, j), (i, -j), (-i, -j), (-j, -i), (-j, i), (j, -i), (j, i)$. Any of these squares are considered threatened.

Input

The first line of the input contains a single positive integer, n, representing the number of boards to analyze. Each board starts with a new line containing two integers, p $(1 \leq p \leq 10^4)$ and k $(1 \leq k \leq 10)$, representing the number of knights and the value k from above, respectively. This is followed by p lines representing each knight. Each of these p lines contains four integers, r, c, a and b $(0 \leq r \leq 10^9; 0 \leq c \leq 10^9; 1 \leq a \leq 10^9; 1 \leq b \leq 10^9)$, representing the row and column the knight occupies as well as its a and b values (from above).

Output

For each board, output the number of squares each of which threatened by k or more knights.

Examples

input	output

3
2 10
10 10 2 1
10 10 1 2
1 1
10 10 2 1
2 2
10 10 2 2
13 10 2 2
	0
	12
	8

Limits

Time limit is 3 seconds.

Memory limit is 256 megabytes.

Table

The new restaurant in town has a large rectangular table where the owner likes to place ornaments. The problem is that such ornaments reduce the amount of flat table surface where waiters can safely place dishes. The restaurant owner wants to make sure that all dishes can be put somewhere on the table in a sufficiently large safe area, that is, without overlapping any ornament. Given the dimensions and locations of all ornamental areas, you must figure out what impact these ornaments have on the locations available for dishes.

The table is a rectangle of width X and length Y, given in millimeters. On top of it, N ornaments are placed. Each of them is located at fixed table coordinates and shaped like a rectangle whose sides are parallel to the sides of the table. Of course, none of these areas overlap each other, but they can come into contact.

All D dishes are rectangular and placed with their sides parallel to the sides of the table, in a predetermined orientation. Waiters have millimetric precision: They will place dishes at integer millimetric coordinates with their sides parallel to those of the table. Dishes should not overlap any ornamental area (but they can touch its edges). Given a list of dishes described by their dimensions, your task is to tell, for each of them, the number of (integer) locations where it can be safely placed on the table. Note: only one dish is served at a time on the table; this means that you do not need to worry about the fact that dishes could overlap, you can count the locations for each dish independently from others.

Input

The input comprises several lines, each consisting of integers separated with single spaces. The first line consists of the integers X, Y, N, and D, with the following limits: $1 \leq X, Y \leq 2000$; $0 \leq N \leq 10^6$; $1 \leq D \leq 10^5$. Each of the N following lines contains the coordinates of an ornament as four integers x, x', y, y', with $0 \leq x < x' \leq X$ and $0 \leq y < y' \leq Y$, describing an ornament spanning between the point (x, y) and the point (x', y'). The D following lines contain the width x and length y of dishes as two integers with $0 < x \leq X$ and $0 < y \leq Y$.

Output

D lines containing the number of valid integer locations for each dish.

Examples

input	output
7 5 3 0	1
1 2 0 1	1
5 7 2 5	0
0 1 2 4	13
7 1	5
3 5	1
5 3	0
2 2	0
3 3	26
4 4	
4 5	
6 2	
1 1	

Limits

Time limit is 3 seconds.

Memory limit is 256 megabytes.