Elevator II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

There is a building with 10^9 floors but only 1 elevator. Initially, the elevator is on the f-th floor.

There are n people waiting for the elevator. The i-th person is currently on the l_i-th floor and wants to take the elevator to the r_i-th floor ($l_i < r_i$). Because the elevator is so small, it can carry at most 1 person at a time.

It costs 1 unit of electric energy to move the elevator 1 floor upwards. No energy is needed if the elevator moves downwards. That is to say, it costs $\max(y - x, 0)$ units of electric energy to move the elevator from the x-th floor to the y-th floor.

Find the optimal order to take all people to their destinations so that the total electric energy cost is minimized.

More formally, let $a_1, a_2, \cdots, a_n$ be a permutation of n where a_i indicates that the i-th person to take the elevator is a_i. The total electric energy cost can be calculated as

$$\sum_{i=1}^{n}(\max(l_{a_i} - r_{a_{(i-1)}}, 0) + r_{a_i} - l_{a_i})$$

where $a_0 = 0, r_0 = f$ for convenience.

Recall that a sequence $a_1, a_2, \cdots, a_n$ of length n is a permutation of n if and only if each integer from 1 to n (both inclusive) appears exactly once in the sequence.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first line contains two integers n and f ($1 \le n \le 10^5$, $1 \le f \le 10^9$) indicating the number of people and the initial position of the elevator.

For the following n lines, the i-th line contains two integers l_i and r_i ($1 \le l_i < r_i \le 10^9$) indicating that the i-th person wants to go from the l_i-th floor to the r_i-th floor by elevator.

It's guaranteed that the sum of n of all test cases will not exceed 3×10^5.

Output

For each test case, first output one line containing one integer indicating the minimum total electric energy, then output another line containing n integers $a_1, a_2, \cdots, a_n$ separated by a space indicating the optimal order to carry all people. Note that these n integers must form a permutation of n. If there are multiple optimal orders, you can print any of them.

Example

standard input	standard output
2 4 2 3 6 1 3 2 7 5 6 2 5 2 4 6 8	11 2 1 4 3 5 2 1