Mahjong Connect

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Busy Beaver has finally rage-quit Minesweeper and picked up Mahjong Connect instead. He has N mahjong tiles at distinct locations on the Cartesian plane. Each tile i has integer coordinates (x_i, y_i) and a type t_i with $1 \le t_i \le M$.

Two distinct tiles i and j match if and only if **all** of the following hold:

- They share the same type, i.e. $t_i = t_j$;
- They are on the same row or column, i.e. $x_i = x_j$ or $y_i = y_j$;
- They are not blocked by other types, i.e. every tile strictly between them in that row or column (if any) has the same type. Formally:
  - If $y_i = y_j$, then for every tile k with $y_k = y_i$ and $\min\{x_i, x_j\} < x_k < \max\{x_i, x_j\}$, we must have $t_k = t_i$;
  - If $x_i = x_j$, then for every tile k with $x_k = x_i$ and $\min\{y_i, y_j\} < y_k < \max\{y_i, y_j\}$, we must have $t_k = t_i$.

A perfect matching is a partition of the N tiles into $N/2$ disjoint pairs such that every pair is a valid match under the rules above. Determine whether a perfect matching exists. If it does, output any one perfect matching; otherwise, report that no perfect matching exists.

Input

The first line contains two integers N and M ($2 \le N \le 3 \cdot 10^5$, $1 \le M \le 3 \cdot 10^5$, N is even).

The next N lines describe the tiles. The i-th of these lines contains three integers x_i, y_i and t_i ($|x_i|, |y_i| \le 10^9$, $1 \le t_i \le M$).

It is guaranteed that the pairs (x_i, y_i) are distinct.

Output

If no perfect matching exists, print a single line containing `NO`.

Otherwise, print a single line containing `YES`. Then print $N/2$ lines, each containing two integers i and j ($1 \le i, j \le N$, $i \ne j$), indicating that tiles i and j form a matched pair.

Each index from 1 to N must appear in exactly one pair. The pairs may be printed in any order, and within each pair, the order of i and j does not matter.

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Examples

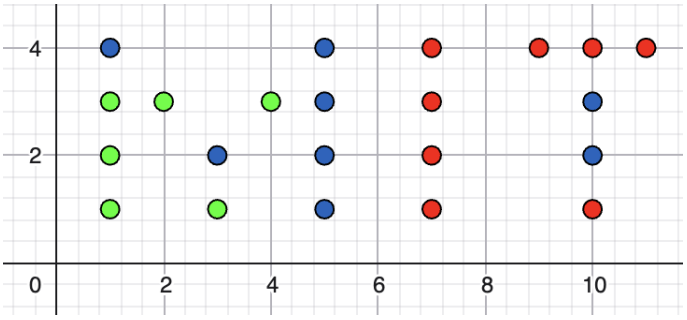
standard input	standard output
4 2 -1 0 1 1 0 1 0 -1 2 0 1 2	YES 1 2 3 4
4 2 -1 0 1 1 0 1 0 0 2 0 1 2	NO
22 3 1 1 2 1 2 2 1 3 2 1 4 1 2 3 2 3 2 1 3 1 2 4 3 2 5 4 1 5 3 1 5 2 1 5 1 1 7 1 3 7 2 3 7 3 3 7 4 3 9 4 3 10 4 3 11 4 3 10 3 1 10 2 1 10 1 3	YES 1 7 2 3 4 9 5 8 6 11 10 12 13 22 14 15 16 17 18 19 20 21

Note

In the first test case, we can pair up the tiles at $(-1, 0), (1, 0)$ and $(0, -1), (0, 1)$, respectively.

In the second test case, we cannot do the same thing because the tile at $(0, 0)$ blocks the connection between $(-1, 0)$ and $(1, 0)$.

The visualization of the third test case is as follows (different colors represent different types of tiles):



Note again that, if the answer is `YES`, **any valid pairing** with **any order** will be accepted. For instance, for the first sample, the following output is also acceptable:

YES
4 3
1 2