ACM-ICPC Live Archive

8275 Annual Congress of MUD

Multiuser dungeon games, also called MUD games, are real-time virtual-world multiplayer games that are usually played online with text-based commands. The players do not meet normally, but every year, there is the *Annual Congress of MUD (ACM)* where MUD lovers all around the world could meet in person.

ACM is so popular that the event each year spans around 20 days. Each day, there will be a special gathering for MUD game designers to introduce their new games to the others. Each player will usually spend a few days on the ACM site, and in-between will be invited in exactly one day to join this special gathering.

This year, ACM is held at your city, and your boss is an organiser, and he wants to find a *best* way to assign the players to these special gatherings (one player to one special gathering within his or her duration of stay), so that the maximum number of players among all gatherings is minimized.

Your boss is an extremely impatient guy. He wants to have a system that can tell the maximum number of players among all gatherings, in any best assignment, after *each* player enters his or her duration of stay. Your task is to help your boss develop such a system.

Technical Specification

1. The number of players joining the congress, N, is an integer with $1 \leq N \leq 10000$.
2. The number of days, D, of the congress is an integer with $3 \leq D \leq 20$.
3. The duration of stay, $[x_i, y_i]$, for the i-th player to be keyed in your system, are pairs of integers with $1 \leq x_i \leq y_i \leq D$. This indicates that the i-th player will stay from day x_i (inclusive) to day y_i (inclusive) on the ACM site.

Input

An instance of the problem consists of $N + 1$ lines. The first line specifies the integers N and D, seperated by a space. In the i-th line of the following N lines, it contains two integers x_i and y_i, separated by a space.

Note that the test data file may contain more than one instance. The last instance is followed by a line containing a single '0'.

Output

For each instance, an integer i is *marked* if and only if the maximum number of players in the best assignment increases after the duration of stay $[x_i, y_i]$ of the i-th player is keyed in. By default, 1 is always marked.

The output of the corresponding instance is the list of all marked integers in ascending order, separated by a space between successive integers, followed by a newline character.

Sample Input

```
3 3
1 1
1 1
1 1
```

```
3 3
1 2
2 3
1 1
3 3
1 2
1 2
1 2
0
```

Sample Output

```
1 2 3
1
1 3
```