

G. Meetings

Hadi loves organizing computer science competitions such as UTCP. Because of this, he often needs to have online meetings with sponsors, professors, volunteers, and so on.

To keep track of all these meetings, he naturally uses a calendar, but the meetings can get so frequent that sometimes more than one meeting is scheduled to take place at the same time! Hadi is keeping up with this for now (virtual meetings, multiple browser windows, nodding and smiling to the camera, ...) but he agrees it should really be avoided as much as possible.

We consider two meetings to happen at the same time even if they just **touch**, for example a meeting taking place from time 1 to time 3 overlaps with another taking place from time 3 to time 6, because William needs some time to switch meetings, he cannot finish one meeting and immediately be connected to the other meeting.

Hadi asked his friend Shayan to implement an algorithm to reduce meeting overlap in his calendar. The calendar has n meetings, the i -th of which starts at time L_i and ends at time R_i . We define the “schedule cost” as the maximum number of meetings that Hadi is attending simultaneously. Shayan’s algorithm is allowed to cancel k meetings and wants to minimize the cost of the new schedule.

Help Shayan write the algorithm for Hadi’s calendar!

Input:

The first line contains two integers: n and k , respectively, the total number of meetings and the maximum number of meetings that Shayan’s algorithm can cancel.

Each of the next n lines contain a pair of integers: L_i and R_i , respectively, the starting and the ending time of the i -th meeting.

Output:

You need to write a single line containing the minimum cost of the new schedule.

Constraints:

- $2 \leq n \leq 10^5$
- $2 \leq L_i < R_i \leq 10^5$
- $1 \leq k < n$
- Each pair (L_i, R_i) is unique.

Sample Test Data

Input	Output
3 1 5 12 2 8 6 15	2

Input	Output
5 2 3 10 6 13 11 19 2 20 4 8	2