A. Rank List (Edited)

time limit per test
2 seconds
memory limit per test
256 megabytes
input
standard input
output
standard output

NO SE PUEDE UTILIZAR EL SORT DE PYTHON

Another programming contest is over. You got hold of the contest's final results table. The table has the following data. For each team we are shown two numbers: the number of problems and the total penalty time. However, for no team we are shown its final place.

You know the rules of comparing the results of two given teams very well. Let's say that team a solved p_a problems with total penalty time t_a and team b solved p_b problems with total penalty time t_b. Team a gets a higher place than team b in the end, if it either solved more problems on the contest, or solved the same number of problems but in less total time. In other words, team a gets a higher place than team b in the final results' table if either $p_a > p_b$, or $p_a = p_b$ and $t_a < t_b$.

It is considered that the teams that solve the same number of problems with the same penalty time share all corresponding places. More formally, let's say there is a group of x teams that solved the same number of problems with the same penalty time. Let's also say that y teams performed better than the teams from this group. In this case all teams from the group share places $y + 1, y + 2, ..., y + x$. The teams that performed worse than the teams from this group, get their places in the results table starting from the $y + x + 1$-th place.

Your task is to count what number of teams from the given list shared the k-th place.

Input

The first line contains two integers n and k ($1 \le k \le n \le 3000$). Then n lines contain the description of the teams: the i-th line contains two integers p_i and t_i ($1 \le p_i, t_i \le 10000$) — the number of solved problems and the total penalty time of the i-th team, correspondingly. All numbers in the lines are separated by spaces.

Output

In the only line print the sought number of teams that got the k-th place in the final results' table.

Examples

input

```
7 2
4 10
4 10
4 10
3 20
2 1
2 1
1 10
```

output

```
3
```

input

```
5 4
3 1
3 1
5 3
3 1
3 1
```

output

```
4
```