Kick Start 2021 - Round D

Cutting Intervals

Problem

You are given $\mathbf{N}$ intervals. An interval can be represented by two positive integers $\mathbf{L_i}$ and $\mathbf{R_i}$ - the interval starts at $\mathbf{L_i}$ and ends at $\mathbf{R_i}$, represented as $[\mathbf{L_i}, \mathbf{R_i}]$. Intervals may not be unique, so there might be multiple intervals with both equal $\mathbf{L_i}$ and equal $\mathbf{R_i}$.

You are allowed to perform a maximum of $\mathbf{C}$ cuts. A cut at X will cut all intervals $[L, R]$ for which $L < X$ and $X < R$. Cutting an interval at X is defined as splitting the interval into two intervals - $[L, X]$ and $[X, R]$. Note that cuts can only be performed at integer points. Also, cutting at an endpoint of an interval ($X = L$ or $X = R$) has no effect and does not split the interval.

You need to find the maximum number of intervals that can be obtained through a maximum of $\mathbf{C}$ cuts.

Input

The first line of the input contains the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow.

Each test case starts with a line containing two integers, $\mathbf{N}$ and $\mathbf{C}$, denoting the number of intervals and the maximum number of cuts you can perform respectively. $\mathbf{N}$ lines follow.
The i-th line contains two integers $\mathbf{L_i}$ and $\mathbf{R_i}$, describing the i-th interval.

Output

For each test case, output one line containing `Case #`x`:` y, where x is the test case number (starting from 1) and y is the maximum number of intervals that can be obtained through at most $\mathbf{C}$ cuts, as described above.

Limits

Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.

Test Set 1

Time limit: 20 seconds.
$1 \leq \mathbf{N} \leq 500$.
$1 \leq \mathbf{C} \leq 10^5$.
$1 \leq \mathbf{L_i} < \mathbf{R_i} \leq 10^4$ for all i.

Test Set 2

Time limit: 40 seconds.
$1 \leq \mathbf{N} \leq 10^5$.
$1 \leq \mathbf{C} \leq 10^{18}$.
$1 \leq \mathbf{L_i} < \mathbf{R_i} \leq 10^{13}$ for all i.

Sample

Sample Input

```
1
3 3
1 3
2 4
1 4
```

Sample Output

```
Case #1: 7
```

In the provided sample, cuts should be performed at 2 and 3 to get the maximum number of intervals.
After the first cut at 2, the intervals would be $\{[1, 2], [2, 3], [2, 4], [1, 2], [2, 4]\}$.
After the second cut at 3, the intervals would be $\{[1, 2], [2, 3], [2, 3], [3, 4], [1, 2], [2, 3], [3, 4]\}$.
It can be seen that no interval can be cut further, so the answer is 7.