Problem G. Gotta Catch 'Em All

Program:	`catcher.(cpp|java)`
Input:	`catcher.in`
Balloon Color:	`Silver`

Ash Ketchum is a Pokémon trainer who is on a mission to catch all Pokémons in the world. Fortunately, he has caught most of them and he needs to collect K more different types of Pokémons to complete his set.

As you may already know, Ash lives in a 2D grid world. Pokémons spawn at integer coordinates on this grid. In this world there can be multiple Pokémons of the same type. Remember, Ash needs to catch K more different types of Pokémons not any K Pokémons.

Ash decided to catch those Pokémons by throwing a big squared net from the sky. A Pokémon is considered to be caught if the Pokémon lies within the boundaries of the net or even on one of its edges. Ash wants to find a square that contains those Pokémons where each side of the square is either parallel to the x-axis or to the y-axis.

Can you help Ash find the minimum side of the square that contains K different types of Pokémons so that he can buy an appropriate sized net to catch'em all? Since nets need to always have a positive area, the side of the square needs to be positive.

Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer **T**, the number of test cases ($1 \leq T \leq 100$).

Each test case starts with a line that contains two space separated integers:

- **N**: Number of Pokémons in the world ($2 \leq N \leq 1000$)
- **K**: Number of types of Pokémons ($2 \leq K \leq 100$)

Followed by **N** lines each containing 3 space separated integers:

- **X**: The x-coordinate of the Pokémon(-1,00,000,000 $\leq X \leq$ 1,00,000,000)
- **Y**: The y-coordinate of the Pokémon (-1,00,000,000 $\leq Y \leq$ 1,00,000,000)
- **Z**: The type of the Pokémon ($1 \leq Z \leq K$)

Output

For each test case, print a single line containing the minimum side of a square that contains **K** different types of Pokémons.

Example

catcher.in	standard output
2 5 2 0 0 1 0 1 1 0 2 1 2 0 2 2 1 2 3 3 0 0 1 1 1 2 2 2 3	2 2