

Code-Eat Switcher

Problem

Umon is a foodie coder. Do you know what two activities that he loves the most? Of course, coding and eating! He always spends the whole day doing only those two activities. However, he thinks that some times of the day are better spent coding, and others are better spent eating.

To illustrate this problem, Umon divides his day into S time slots. During the i -th time slot, if Umon codes 100% of the time, he will achieve C_i units of coding. On the other hand, if he eats 100% of the time, he will achieve E_i units of eating. But of course, Umon can also use only a fraction of the time for coding, and the remaining for eating. Formally, he will choose a real number f ($0 \leq f \leq 1$), code for f of the time, and use the remaining $(1 - f)$ time to eat. This way, he will achieve $f \times C_i$ units of coding and $(1 - f) \times E_i$ units of eating. The total amount of coding Umon achieves for the day is simply the sum of all units of coding he achieved in each of the time slots. The total amount of eating is calculated in a similar way.

Umon needs to plan his schedule for the next D days. On the i -th day, he needs to achieve at least a total amount of A_i units of coding and B_i units of eating. For each day, determine whether there is a way for Umon to achieve his target.

Input

The first line of input gives the number of test cases, T . T test cases follow. Each test case begins with a line containing two integers D and S , the number of days and the number of time slots in a day, respectively.

Then S lines follow, each describing a time slot. The i -th line contains two integers C_i and E_i , the amount of coding units achieved if Umon codes for 100% of the time slot, and the amount of eating units achieved if he eats for 100% of the time slot, respectively.

Then D lines follow, each describing a day. The i -th line contains two integers A_i and B_i , the minimal total amount of coding and eating that needs to be achieved on that day.

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 a string with D characters, where the i -th character is Y if there exists a schedule that can fulfill the target for the i -th day, otherwise it should be N .

Limits

Time limit: 20 seconds per test set.

Memory limit: 1GB.

$1 \leq T \leq 100$.

$1 \leq C_i \leq 10^4$, for all i .

$1 \leq E_i \leq 10^4$, for all i .

$0 \leq A_i \leq 10^8$, for all i .

$0 \leq B_i \leq 10^8$, for all i .

Test set 1 (Visible)

$$1 \leq \mathbf{S} \leq 2.$$
$$1 \leq \mathbf{D} \leq 10.$$

Test set 2 (Hidden)

For at least TODO% of the test cases:

$$1 \leq \mathbf{S} \leq 10^3.$$
$$1 \leq \mathbf{D} \leq 10^3.$$

For all test cases:

$$1 \leq \mathbf{S} \leq 10^5.$$
$$1 \leq \mathbf{D} \leq 10^5.$$

Sample

Sample Input	Sample Output
2 4 2 3 8 6 10 0 18 3 13 10 0 7 3 1 2 4 4 4 4 0 0	Case #1: YYNY Case #2: Y

In the first sample case, there are 4 days and 2 time slots for each day.

- For day 1, Umon can just eat 100% for both time slots, and therefore achieving a total of 0 units of coding and $8 + 10 = 18$ units of eating, thus reaching the target.
- For day 2, Umon can eat 100% of the time for the first time slot, and use 50% of the second time slot for coding and 50% for eating, achieving a total of $0 \times 3 + 0.5 \times 6 = 3$ units of coding, and $1 \times 8 + 0.5 \times 10 = 13$ units of eating, thus reaching the target.
- For day 3, it is impossible to get a total of 10 units of coding.
- For day 4, there are an infinite amount of ways to achieve the target. One possible strategy is to code 42% (and eat 58%) in the first time slot, then code 98.76% (and eat 1.24%) in the second time slot. That strategy yields a total of $0.42 \times 3 + 0.9876 \times 6 = 7.1856$ units of coding, and $0.58 \times 8 + 0.0124 \times 10 = 4.764$ units of eating.

Thus, the answer should be YYNY.

In the second sample case, note that the value of characteristics for the time slots may not necessarily be different from each other.