

Poker

As you know by now, Lea has always been interested both in games and statistics. Lately, she came to realize that there is no game as good as combining these two as Poker. She began reading up on strategy - soon things like “short stack”, “big stack”, “sharks” etc. began making sense to her. Her iron will to win awakened as she delved headfirst into the world of competitive poker.

She decided she would make a grand entrance to the scene by winning as much money as she could in a single month. This would be easy - she expected to be able to win all amateur tournaments easily with her newfound knowledge. Today, she had a look at the timetable for all the different amateur poker tournaments that would take place over the next months. Unfortunately, some of them overlapped, so she would not be able to win all of them. Can you tell her how much money she can win without going to overlapping tournaments?

Input

The first line of the input contains an integer t . t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing n , the amount of tournaments that take place in the next few months. n lines follow, one for each tournament. Each line consists of 3 numbers, a b p , where the tournament starts on day a , ends on day b and has a prize of p € for the winner. Two tournaments overlap even if the first one ends on the same day the second one starts.

Output

For each test case, print a line containing “Case # i : x ” where i is its number, starting at 1 and x is the most amount of money Lea can win from all the tournaments (given that she wins each tournament she attends). Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 5000$
- $1 \leq a \leq b \leq 5000$
- $1 \leq p \leq 10000$

Sample Input 1

```
2
3
1 5 1000
10 15 1000
2 12 3000

4
1 1 500
2 5 500
3 7 1000
6 7 600
```

Sample Output 1

```
Case #1: 3000
Case #2: 1600
```

Sample Input 2

```
5
3
6 9 10
9 13 8
10 15 8
```

```
4
3 8 6
7 11 10
5 9 10
8 11 9
```

```
3
9 12 5
8 13 8
1 6 6
```

```
3
8 11 10
1 5 6
5 10 11
```

```
4
2 7 7
4 7 8
3 8 5
1 6 6
```

Sample Output 2

```
Case #1: 18
Case #2: 10
Case #3: 14
Case #4: 16
Case #5: 8
```