

There is a very large square room of size 10000×10000 . In this room there are several square shaped compartments. Each compartment has a maximum area of 10000 square units. The compartments are situated in random positions with the walls parallel to the walls of the room. Some of the compartments overlap each other. Every compartment is strictly inside the room (ie. There are no walls that are common to both the room and the compartments).

There is a predator somewhere in the room. The predator can not walk through the walls of the compartment or the room. You have to find out the area that could be reached by the predator.

X	1	2	3	4	5	6	7	8	9	10
1										
2										
3		+								
4										
5										
6										
7										
8										
9										
10										

The room could be modeled as a square grid of size 10000×10000 .

The figure shows a part of the room. We can define each compartment by 3 integers. It will be in the order (r, c, s) . Where r and c indicates the upper left corner of the square and s determines its size. Note that the sides of the square passes through the grid lines. So in the figure above, the red square has the format $(2, 2, 3)$, the blue square has the format $(6, 3, 4)$ and the green square has the format $(4, 6, 3)$.

Input

There will be multiple test cases. Each case starts with an integer C ($C \leq 4$) that determines the number of compartments in the room. The next C lines will contain three integers each that will give the location of the compartments in the format (r, c, s) . The next line will contain an integer Q ($Q \leq 5$). Each of the next Q lines will contain two integers pr pc that will give the location of the predator.

Note: pr is the row and pc is the column of the predators's position in the grid. The '+' sign on the grid has (pr, pc) value of $(3, 2)$.

Output

For each test case, output the case number. Then for each query, output an integer that will give the area that the predator can cover.

Look at the sample for exact format.

Sample Input

```
3
2 2 3
6 3 4
4 6 3
3
3 2
6 6
10000 10000
1
5 5 10
1
5 5
```

Sample Output

```
Case 1:
9
1
99999967
Case 2:
100
```