

## 1266 – Points in Rectangle

As the name says, this problem is about finding the number of points in a rectangle whose sides are parallel to axis. All the points and rectangles consist of 2D Cartesian co-ordinates. A point that lies in the boundary of a rectangle is considered inside.

### Input

Input starts with an integer **T** ( $\leq 10$ ), denoting the number of test cases.

Each case starts with a line containing an integer **q** ( $1 \leq q \leq 30000$ ) denoting the number of queries.

Each query is either one of the following:

- 1) **0 x y**, meaning that you have got a new point whose co-ordinate is **(x, y)**. But the restriction is that, if a point **(x, y)** is already listed, then this query has no effect.
- 2) **1 x<sub>1</sub> y<sub>1</sub> x<sub>2</sub> y<sub>2</sub>** meaning that you are given a rectangle whose lower left co-ordinate is **(x<sub>1</sub>, y<sub>1</sub>)** and upper-right corner is **(x<sub>2</sub>, y<sub>2</sub>)**; your task is to find the number of points, given so far, that lie inside this rectangle. You can assume that **(x<sub>1</sub> < x<sub>2</sub>, y<sub>1</sub> < y<sub>2</sub>)**.

You can assume that the values of the co-ordinates lie between **0** and **1000** (inclusive).

### Output

For each case, print the case number in a line first. Then for each query type (2), you have to answer the number of points that lie inside that rectangle. Print each of the results in separated lines.

Sample Input	Output for Sample Input
1 9 0 1 1 0 2 6 1 1 1 6 6 1 2 2 5 5 0 5 5 1 0 0 6 5 0 3 3 0 2 6 1 2 1 10 10	Case 1 : 2 0 2 3

### Note

Dataset is huge, use faster I/O methods.