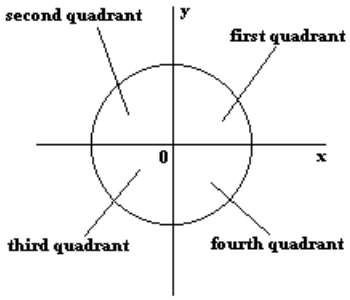
Quadrant Queries

HackerRank

There are n points on a plane. Each point $p[i]$ is described by $[x[i], y[i]]$, where $1 \leq i \leq n$. There are three types of queries needed:

1. `X i j` Reflect all points in the inclusive range between points $p[i]$ and $p[j]$ along the x-axis.
2. `Y i j` Reflect all points in the inclusive range between points $p[i]$ and $p[j]$ along the y-axis.
3. `C i j` Count the number of points in the inclusive range between points $p[i]$ and $p[j]$ in each of the 4 quadrants. Then print a single line of four space-separated integers describing the respective numbers of points in the first, second, third, and fourth quadrants in that order.

As a reminder, the four quadrants of a graph are labeled as follows:



Given a set of n points and q queries, perform each query in order. For example, given points $p = [(1,1),(-1,-1)]$ and $queries = [\texttt{'X 1 2', 'C 1 2', 'Y 1 1' 'C 1 2'}]$. Initially the points are in quadrants 1 and 3. The first query says to reflect points with indices from 1 to 2 along the x-axis. After the query, $p = [(1,-1),(-1,1)]$ and quadrants are 4 and 2. The next query prints the number of points in each quadrant: `0 1 0 1`. The third query says to reflect the point with index 1 to 1 along the y-axis, so now $p = [(-1,-1),(-1,1)]$. The points now lie in quadrants 3 and 2, so the fourth query output is `0 1 1 0`.

Note: Points may sometimes share the same coordinates.

Function Description

Complete the *quadrants* function in the editor below. It should print the results of each `C` type query on a new line.

quadrants has the following parameters:
- *p[p[1]...p[n]]*: a 2-dimensional array of integers where each element $p[i]$ contains two integers $x[i]$ and $y[i]$
- *queries[queries[1]...queries[n]*: an array of strings

Input Format

The first line contains a single integer, n, that denotes the number of points.
Each line i of the n subsequent lines contains two space-separated integers that describe the respective

$x[i]$ and $y[i]$ values for point $p[i]$.
The next line contains a single integer, q, that denotes the number of queries.
Each of the q subsequent lines contains three space-separated values that describe a query in one of the three forms defined above.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq q \leq 10^6$
- No point lies on the x or y axes.
- $1 \leq x[i], y[i] \leq 2^{31} - 1$
- In all queries, $1 \leq i \leq j \leq n$.

Output Format

For each query of type `C i j`, print four space-separated integers that describe the number of points having indices in the inclusive range between i and j in the first, second, third, and fourth graph quadrants in that order.

Sample Input

```
4
1 1
-1 1
-1 -1
1 -1
5
C 1 4
X 2 4
C 3 4
Y 1 2
C 1 3
```

Sample Output

```
1 1 1 1
1 1 0 0
0 2 0 1
```

Explanation

Initially, $p = [[1, 1], [-1, 1], [-1, -1], [1, -1]]$ so there is one point in each of the four quadrants. The first query results in printing `1 1 1 1`.

The second query, `X 2 4`, reflects the points in the inclusive range between indices 2 and 4 along the x-axis. Now $p = [[1, 1], [-1, -1], [-1, 1], [1, 1]]$.

The query `C 3 4` requires that the number of points considering $p[3]$ through $p[4]$ be printed: `1 1 0 0`

The third query, `Y 1 2` requires reflection of $p[1] - p[2]$ along the y-axis. Now $p = [[-1, 1], [1, -1], [-1, 1], [1, 1]]$.

The last query, `C 1 3` requires that the number of points considering $p[1]$ through $p[3]$ be printed: `0 2 0 1`