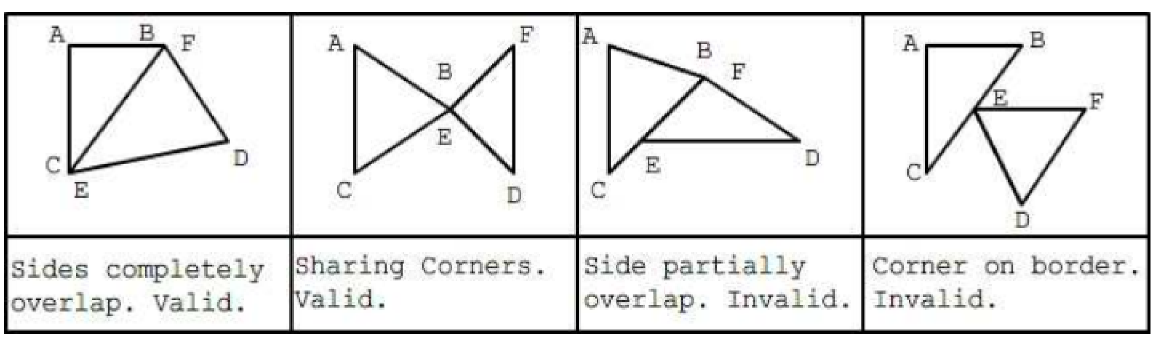
Have you ever been to safari park? It is a zoo-like place where animals are not kept under cage. Visitors may drive or walk through safari and observe free animals. Even sometimes dangerous animal like lion, tiger are also kept free. I have also opened a safari park of my own. Animals live in different regions here. Region of one animal does not intersect with regions of other animal for safety purpose. But two regions may have same boundary though it is not allowed to have the corner point on any other boundary. The animals are trained so that they do not leave their own territory. Now the safari park is so big that a single map cannot show the whole place. So I decided to make a device to help people to find the places of their interest. However, to reduce the cost for device I bought some low end one. Their memory is quite low. To make things less complicated and efficient from processing perspective, we have made all the regions triangle. But that is not all. All the regions will not be loaded initially. Regions will be loaded depending on the visitor's place. Visitor may visit from one place to another place park by mono-rail. So the triangles may appear in the device in different places. Since the device is not that much advance, it can show the regions but it does not show any label. One has to enter the co-ordinate of a point in a text box and the device will show the label of that region in an alert message. There are two special labels. '`0`' denotes outside of any region, '`-1`' denotes on boundary / corner.

For example, suppose currently we know about two regions: Giraffe: (0, 0), (0, 5), (5, 0) and Dolphin: (10, 10), (10, 5), (5, 10). Now if the query co-ordinate is: (1, 1) then it will show Giraffe. If the query is for (9, 9) the answer will be Dolphin. However if the query is: (5, 5) it will show 0. Now suppose a new region appears in the map: Lion: (4, 6), (6, 4), (4, 4). Now again if the query is (5, 5) the answer will be '`-1`'.

So to summarize, you will be given some triangles. Triangle may share side and also corners, but no triangle will have its corner on any sides of any other triangle, i.e. sides of different triangles can be identical, but not partially overlapping. You may check the following diagram for clarity:



Sides completely overlap. Valid.	Sharing Corners. Valid.	Side partially overlap. Invalid.	Corner on border. Invalid.

No two triangles will have common region. If a query point is strictly inside some region you should print its label, if it is strictly outside all regions you should print '`0`' and if it is on some boundary / corner you should print '`-1`'. Details of input and output can be found in corresponding section.

Input

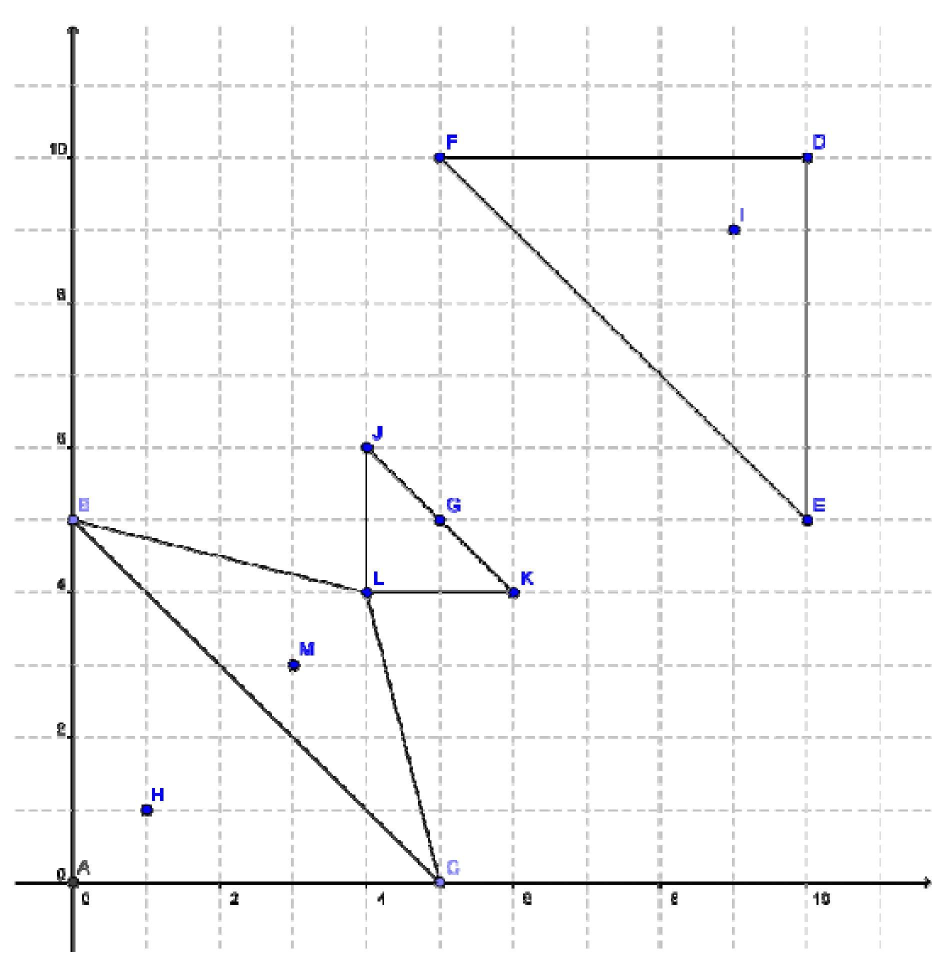
First line of the test file contains a positive integer T (≤ 2) which is the number of test cases. First line of each case contains N ($\leq 300,000$) denotes number of commands. Each command will start with a '`Q`' or '`R`'. `Q` stands for query and `R` stands for region. `Q` will be followed by 2 integers: x_q and y_q. `R` will be followed by 6 integers: x_1 y_1 x_2 y_2 x_3 y_3. You may assume that there won't be more than 50,000 `R` commands.

However these numbers will not give you the real co-ordinate. You have to decode the given coordinates using the answer to last query. Suppose d is the answer to the last query `Q` (Initially $d = 0$), then real co-ordinate of given co-ordinate x, y will be: $x + x_1[d]$, $y + y_1[d]$. Here, $x_1[d]$, $y_1[d]$ is the first co-ordinate (decoded) of d-th region. d-th region is the region provided by d-th `R` command. However, if d = `0` or `-1` (in case of out of region or on boundary / corner) you should consider $x_1[d] = y_1[d] = 0$. A query should be answered considering only the regions provided before the corresponding `Q` command. If there are n `R` commands before a `Q` command then answer to this command will range from `-1` to n, '`-1`' if on boundary / corner, '`0`' if outside of any region and other number depending on the region in which the point is in. You may assume that the co-ordinates of the regions will always be in clock wise order. You may also assume that the regions will appear in completely random order even though the regions may not be random themselves. Value of decoded x y co-ordinates are non negative and will be bounded by 100,000.

Output

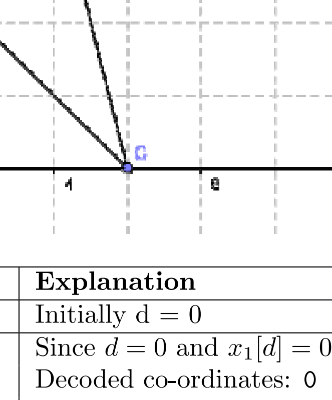
For each test case first print case number. Hence for each `Q` command print the answer of the query. Do not print any blank line between test cases. For details follow the sample input output.

Explanation of sample input:



Sample input	Explanation
8	Initially d = 0
`R 0 0 0 5 5 0`	Since $d = 0$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `0 0 0 5 5 0` In the picture it is ABC triangle
`R 10 10 10 5 5 10`	Since $d = 0$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `10 10 10 5 5 10` In the picture it is DEF triangle
`Q 5 5`	Since $d = 0$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `5 5` This is out of region. Answer is `0` and so $d = 0$. In the picture it is G point
`Q 1 1`	Since $d = 0$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `1 1` This is in 1st region. Answer is `1` and so $d = 1$. In the picture it is H point
`Q 9 9`	Since $d = 1$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `9 9` This is in 2nd region. Answer is `2` and so $d = 2$. In the picture it is I point
`R -6 -4 -4 -6 -6 -6`	Since $d = 2$ and $x_1[d] = 10$, $y_1[d] = 10$ Decoded co-ordinates: `4 6 6 4 4 4` In the picture it is JKL triangle
`Q -5 -5`	Since $d = 2$ and $x_1[d] = 10$, $y_1[d] = 10$ Decoded co-ordinates: `5 5` This is on the boundary of 3rd region. Answer is `-1` and so $d = -1$. In the picture it is G point again
`Q 1 1`	Since $d = -1$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `1 1` This is in 1st region. Answer is `1` and so $d = 1$. In the picture it is H point again
`R 0 5 4 4 5 0`	Since $d = 1$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `0 5 4 4 5 0` In the picture it is BLC triangle
`Q 3 3`	Since $d = 1$ and $x_1[d] = 0$, $y_1[d] = 0$ Decoded co-ordinates: `3 3` This is in 4th region. Answer is `4` and so $d = 4$. In the picture it is M point

The following picture should give you an idea of how one of the judge test case look like:



Sample Input

```
1
10
R 0 0 0 5 5 0
R 10 10 10 5 5 10
Q 5 5
Q 1 1
Q 9 9
R -6 -4 -4 -6 -6 -6
Q -5 -5
Q 1 1
R 0 5 4 4 5 0
Q 3 3
```

Sample Output

```
Case 1:
0
1
2
-1
1
4
```