Transforms

For a lot of graphical applications, such as video games, 3d-modeling software, etc. it is imperative to be able to take a point in space and transform it to a different location. For this problem, you will be given a large array of transformations that can be applied to a point in 2d space. Each transformation will either be a translation (move the point), rotation (around the origin), or scale (increase distance from origin by some factor). Given this large list of transformations, you will write a program to answer queries of the following form: Given an (x, y) coordinate in space, a left index l, and a right index r, what is the new location of this point after applying all of the transformations in the array from index l to r to this point?

The three transformations that we will use in this problem are specified below:

- **Translation:** Move the point (x, y) by some amount (dx, dy), where dx and dy are the amount to move the point in the x and y directions respectively.
- **Scale:** Move the point away from the origin by some multiplicative factor (sx, sy) (this is useful for zooming in on objects / scenes, etc.).
- **Rotation:** Rotate the point around the origin by some number of degrees d. Note that positive rotations should move counterclockwise around the origin.

Input

The first line of input will be the number of transformations $n \leq 10^5$ and the number of queries $q \leq 100000$. The next n lines will provide a transformation to store initially in each of the list of n transformations in order. Each of these will either say "Translate dx dy", "Scale sx sy", or "Rotate d".

The next q lines will each have a query in one of the following formats:

- Q (for query) following by the initial point location x,y following by the range of transformations in your list to apply l and r. For example, "Q 5 5 3 8" is requesting that the point $(5, 5)$ be passed through the transformations in the list from index 3 to 8 inclusive.
- U (for update) followed by an index i followed by a new transformation in the same format as the original transformations above. You should replace the transformation at index i with the new transformation.

Note that translation, scaling, and rotation values may be floating point numbers (even though in the sample below they are not)

Output

For each query in the input, output the original point location in the format (x, y) : followed by one space, followed by the new location $x'y'$ after the transformations in question have been applied. The new point should be space separated as in the sample below. **You should report your answers such that the absolute or relative error is no more than 10^{-5}.**

A Note on Language and Efficiency

We highly recommend using *c++* for this assignment if you want full credit. While implementations (even somewhat optimized naive solutions) can pass many of the smaller cases, we did not produce a working *python* solution that passes all of the test cases. We were able to produce multiple *c++* solutions that do though. The time limit for each test case is 3 seconds, which is pretty lenient for fast highly optimized solutions. A c++ solution that is implemented as described in class should run within this time limit. You have been warned.

Sample Input

```
3 3
Translate 0 2
Scale 0 3
Rotate 90
Q 0 0 0 2
U 2 Translate 5.5 0
Q 2 2 1 2
```

Sample Output

```
(0,0):  -6.0 0.00
(2,2):  5.5 6
```