

# Resizable and moving rectangle

Create a class named **Rectangle**. You may create other classes if needed to support this.

It should provide a way to initialize its state according to given points. And initialization should be allowed only once.

It should provide a way to increase/decrease its height from top and bottom by given units.

It should provide a way to increase/decrease its width from left and right by given units.

It should provide a way to move whole rectangle to left, right, up and down by given units.

NOTE: We are only concerned with rectangle whose sides are parallel to either x-axis or y-axis. You do not need to worry about rectangles whose sides are not parallel to axes. If you are done with this problem and if you are interested to play with it then you can extend capabilities of your class to handle rectangles whose sides are not parallel to the axes

## Input Format

First four lines contain two numbers each (separated by space) representing the co-ordinates of four corners of the rectangle.

First line represents the top-left corner

Second line represents the top-right corner

Third line represent the bottom-left corner

Fourth line represents the bottom-right corner

Fifth line contains number  $n$

Next  $n$  lines contain commands to move/resize the rectangle.

If first letter of line is (L, R, U or D), then rectangle needs to move in respective direction (Left, Right, Up or Down) by units specified in the same line.

If first letter of line is A, then it means alter the size of rectangle. Letter A would be followed by space and then letter I or D - I means increase length/width and D means decrease length or width. I or D would be followed by space and then one of the letter L, R, T or B (representing Left, Right, Top, Bottom resp.). And these L, R, T, B represent the direction in which size of the rectangle should be altered. L and R represent alteration of length from left and right side resp. While T and B represent alteration of width from top or bottom resp.

## Constraints

$1 \leq n \leq 10000$

x and y co-ordinates of all corners will always remain in -100000 to 100000 range

## Output Format

Output should contain four lines representing four corners of final rectangle

First line should represent the top-left corner

Second line should represent the top-right corner

Third line should represent the bottom-left corner

Fourth line should represent the bottom-right corner

### Sample Input 0

```
0 10
20 10
0 0
20 0
1
L 10
```

### Sample Output 0

```
-10 10
10 10
-10 0
10 0
```

### Sample Input 1

```
0 10
20 10
0 0
20 0
1
D 10
```

### Sample Output 1

```
0 0
20 0
0 -10
20 -10
```

### Sample Input 2

```
0 10
20 10
0 0
20 0
2
R 10
U 10
```

### Sample Output 2

```
10 20
30 20
10 10
30 10
```

### Sample Input 3

```
0 10
20 10
0 0
20 0
4
R 10
U 10
A I T 5
A D L 5
```

### Sample Output 3

```
15 25
30 25
15 10
30 10
```