COCI '11 Contest 3 #4 Robot

Time limit: 1.0s **Memory limit:** 32M

Mirko created a new robot and decided to test it on a giant test track. We can imagine the test track as 2D coordinate system. The robot **starts at a point** $(0, 0)$ and receives a set of instructions denoted by letters S, J, I, Z, each of them marking a direction in which robot should be moving.

More precisely, if a robot is located in (x, y), `S` (north) means it should move to $(x, y + 1)$, `J` (south) means it should move to $(x, y - 1)$, `I` (east) means it should move to $(x + 1, y)$ and `Z` (west) means it should move to $(x - 1, y)$.

While robot is receiving instructions and moves through the test track, Mirko is verifying its position in the following manner. Test track contains N fixed **control points**. After each instruction is made, each of the control points measures **Manhattan-distance** to the robot. Distances from all control points are then summed and sent to Mirko.

Assuming that robot moves by the instructions without error, calculate the sum of distances to all control points after each instruction.

Remark: Manhattan-distance of the points (x_1, y_1) and (x_2, y_2) is equal to $|x_1 - x_2| + |y_1 - y_2|$.

Input Specification

First line of input contains positive integers N (number of control points, $1 \leq N \leq 100\,000$) and M (number of instructions, $1 \leq M \leq 300\,000$), separated by a single space.

Each of the following N lines contains coordinates of one control point: two space-separated integers x, y, with absolute value less than $1\,000\,000$ (million). It is possible that two control points have the same coordinates - distance towards each of them is added to the sum.

The following line contains a string of M characters from the set $\{S, J, I, Z\}$, the sequence of instructions sent to the robot.

Output Specification

Output M lines: i^{th} line of output should contain the described number after i^{th} instruction.

Sample Input 1

```
1 3
0 -10
ISI
```

Sample Output 1

```
11
12
13
```

Sample Input 2

```
3 5
0 0
1 1
1 -1
SIJJZ
```

Sample Output 2

```
5
4
3
4
5
```