6836 Automotive Navigation

The International Commission for Perfect Cars (ICPC) has constructed a city scale test course for advanced driver assistance systems. Your company, namely the Automotive Control Machines (ACM), is appointed by ICPC to make test runs on the course.

The test course consists of streets, each running straight either east-west or north-south. No streets of the test course have dead ends, that is, at each end of a street, it meets another one. There are no grade separated streets either, and so if a pair of orthogonal streets run through the same geographical location, they always meet at a crossing or a junction, where a car can turn from one to the other. No U-turns are allowed on the test course and a car never moves outside of the streets.

Oops! You have just received an error report telling that the GPS (Global Positioning System) unit of a car running on the test course was broken and the driver got lost. Fortunately, however, the odometer and the electronic compass of the car are still alive.

You are requested to write a program to estimate the current location of the car from available information. You have the car's location just before its GPS unit was broken. Also, you can remotely measure the running distance and the direction of the car once every time unit. The measured direction of the car is one of north, east, south, and west. If you measure the direction of the car while it is making a turn, the measurement result can be the direction either before or after the turn. You can assume that the width of each street is zero.

The car's direction when the GPS unit was broken is not known. You should consider every possible direction consistent with the street on which the car was running at that time.

Input

The input file contains several test cases, each of them as described below.

The first line contains four integers n, x_0, y_0, t, which are the number of streets ($4 \leq n \leq 50$), x- and y-coordinates of the car at time zero, when the GPS unit was broken, and the current time ($1 \leq t \leq 100$), respectively. (x_0, y_0) is of course on some street. This is followed by n lines, each containing four integers x_s, y_s, x_e, y_e describing a street from (x_s, y_s) to (x_e, y_e) where $(x_s, y_s) \neq (x_e, y_e)$. Since each street runs either east-west or north-south, $x_s = x_e$ or $y_s = y_e$ is also satisfied. You can assume that no two parallel streets overlap or meet. In this coordinate system, the x- and y-axes point east and north, respectively. Each input coordinate is non-negative and at most 50. Each of the remaining t lines contains an integer d_i ($1 \leq d_i \leq 10$), specifying the measured running distance from time $i-1$ to i, and a letter c_i, denoting the measured direction of the car at time i and being either '`N`' for north, '`E`' for east, '`W`' for west, or '`S`' for south.

Output

For each test case, output all the possible current locations of the car that are consistent with the measurements. If they are (x_1, y_1), (x_2, y_2), ..., (x_p, y_p) in the lexicographic order, that is, $x_i < x_j$ or $x_i = x_j$ and $y_i < y_j$ if $1 \leq i < j \leq p$, output the following:

$x1$ $y1$
x_2 y_2
$\vdots$
x_p y_p

Each output line should consist of two integers separated by a space.

You can assume that at least one location on a street is consistent with the measurements.

Sample Input

```
4 2 1 1
1 1 1 2
2 2 2 1
2 2 1 2
1 1 2 1
9 N
6 0 0 2
0 0 2 0
0 1 2 1
0 2 2 2
0 0 0 2
1 0 1 2
2 0 2 2
2 E
7 N
7 10 0 1
5 0 10 0
8 5 15 5
5 10 15 10
5 0 5 10
8 5 8 10
10 0 10 10
15 5 15 10
10 N
```

Sample Output

```
1 1
2 2
0 1
1 0
1 2
2 1
5 5
8 8
10 10
15 5
```