IV BXComp

4th Freshmen's Programming Championship of Information Systems 2014

5th Stage – Challenge 2

Pokémon

Have you ever run into a Pokémon? That is exactly what happens when you walk through the grasses of Kanto because they are full of different Pokémon.

Ash is your friend and he wants to be a Pokémon Master. That means he needs to catch one of each different Pokémon. For that big task, he needs to walk through the whole Kanto, in order to search for Pokémon in grasses. Ash knows you are a good programmer, so he asked you to build a computer program to help him find the number of Pokémon that he can run into during a given period of time.

Task

Your task is quite simple. Given Ash's initial position, speed and movement direction, the number of Pokémon on that scenario, the period of time, and each Pokémon's initial position, speed and movement direction, your program should print the answer to this question: "If Ash keeps moving in that direction, in the next **T** units of time, how many Pokémon would he find?"

Input

The first line of the input file consists of the number **n** of test cases that will be given.

For each test case, in the first line we have:

- **X_A**, **Y_A**, **V_A** and **D_A**, separated by one empty space each, meaning X and Y coordinates of Ash's initial position, the value of his speed and his movement direction, respectively.

Also, for each test case, in the second line we have two integers:

- The former is the number of Pokémon **N_P**;
- The latter is the time **T** that you will consider during this test case.

The next **N_P** lines, in each test case, have the description of each Pokémon:

- **X_P**, **Y_P**, **V_P** and **D_P**, separated by one empty space each, meaning X and Y coordinates of the Pokémon's initial position, the value of its speed and its movement direction, respectively.

There is a blank line after each test case.

$\mathbf{D_A}$ and $\mathbf{D_P}$ can assume values as follows: *N*, *S*, *E*, *W*, *NE*, *NW*, *SE* or *SW,* meaning, respectively, North, South, East, West, Northeast, Northwest, Southeast and Southwest.

Output

For each test case, your program should print a line containg one of the following phrases: “NO POKEMON IN THIS DIRECTION :(“ if Ash did not run into any Pokémon moving in that direction, or “**N** POKEMON IN THIS DIRECTION!” where **N** is the number of Pokémon Ash runs into that way.

Input Example

```
2
0 0 1 N
5 10
0 2 1 S
-1 1 1 E
3 3 2 SW
1 3 1 SE
1 1 3 N

7 0 1 S
2 10
5 2 1 S
1 1 1 E

```

Output Example

```
2 POKEMON IN THIS DIRECTION!
NO POKEMON IN THIS DIRECTION :(
```

Observations

- When Ash’s and the Pokémon’s directions are opposite, $\mathbf{X_A}$ and $\mathbf{X_P}$ will always be even or odd at the same time. The same occours for $\mathbf{Y_A}$ and $\mathbf{Y_P}$ when their directions are opposite.
- There are no test cases where Ash’s and one of the Pokémon’s values for **X**, **Y**, **V** and **D** will be exactly equal.
- There are no test cases where Ash and one of the Pokémon are on the same point at time **T** = 0.