

# A - Round About

**Input:** standard input

**Output:** standard output

The purpose of a roundabout is to control the flow of traffic at a busy intersection. A roundabout has 4 entry points: from the North, East, South and West; and 4 exit points at the same locations.

Cars can enter, leave or move one position on the roundabout. Each of those events requires one second. All the events can take place simultaneously. So if there are two cars on the roundabout, one behind the other, then they can both move during the next second. A car decides whether to enter the roundabout based on its knowledge from the previous second. A car will always enter if it has the right to do so.

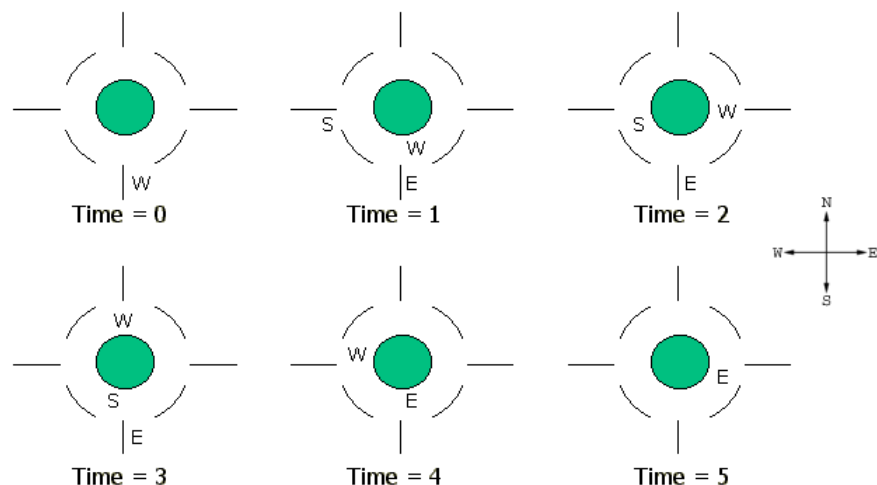
A car has the right to enter only if there are no cars to its immediate left (either on the roundabout or waiting to get on the roundabout) and it is the first car in line at the entry point. A special case occurs if there is a car at each of the 4 entry points. In this case, the car from the North will wait until there are no cars to its left on the roundabout and then will be the first to enter.

Once the car is on the roundabout, it moves counter-clockwise until it exits the roundabout. There are 4 positions on the roundabout. So for example, it would take 1 second to enter the roundabout, 2 seconds to complete half a lap and another 1 second to exit the roundabout.

Each parameter of the input is a second by second account of cars coming from that direction with their intended exit locations. Intended exit locations will be N (North), E (East), S (South) and W (West). A '-' means that no car arrived during that second at the given entry point. Thus, for example, character  $i$  of north represents the direction in which a car (if there is one) arriving from the north at time  $i$  will leave the roundabout. So, at time  $i$ , this car will be added to the end of the line at the north entry point to the roundabout. Cars will not be allowed to exit the roundabout at the point of their entry.

Given the lists of cars coming from the 4 directions, print one line with the total time required for all cars to leave the roundabout.

See the next example for a diagram:



At the start, a car approaches the roundabout from the South. At time = 1, the car going West enters the roundabout, another car approaches from the South and another one from the West. At time = 2, the car going West moves one position, the car going East must give way, the car going South enters the roundabout. At time = 3, the car going West moves one position, the car going East must still give way, the car going South moves one position. At time = 4, the car going West moves one position, the car going South exits the roundabout, the car going East enters the roundabout. At time = 5, the car going West exits the roundabout, the car going East moves one position. At time = 6, the car going East exits the roundabout. The answer should be 6.

## Input

On the first line one positive number: the number of testcases. After that per testcase:

- One line with a string *north* specifying the list of cars coming from the north.
- One line with a string *east* specifying the list of cars coming from the east.
- One line with a string *south* specifying the list of cars coming from the south.
- One line with a string *west* specifying the list of cars coming from the west.

Constraints:

- *north*, *east*, *south* and *west* will contain between 0 and 50 characters inclusive.
- *north*, *east*, *south* and *west* must only contain 'N', 'W', 'S', 'E' and '-' characters.
- *north* will NOT contain the character 'N'.
- *east* will NOT contain the character 'E'.
- *south* will NOT contain the character 'S'.
- *west* will NOT contain the character 'W'.

## Output

Per testcase print one line with the total time required for all cars to leave the roundabout.

## Sample Input

```
7
--
--
WE
-S
WWW
NNN
---
```

```

---
SSS
WW-
N
S-----
SSS-
--W---W
WE
-S
E
-N
W
-S
ES
N
E

E
SN
-N
S-E

```

## Sample Output

```

6
9
13
14
5
9
12

```

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2006, TopCoder, Inc. All rights reserved.

*You may feel free to use old TopCoder algorithm problems for non-commercial, educational purposes. Just be sure to carry over the copyright notice in any reproduction of the problem statement.*