

Exam "Fundamentals of programming with C#" – 27 May 2018

Problem 2. Compass

1. History

The coders in the World of Codecraft are mighty and strong. However, they suck at geography. They cannot operate a compass and they need your help to find the beasts they need to fight. **Your task will be to help our heroes and provide them with some directions.**

2. Description

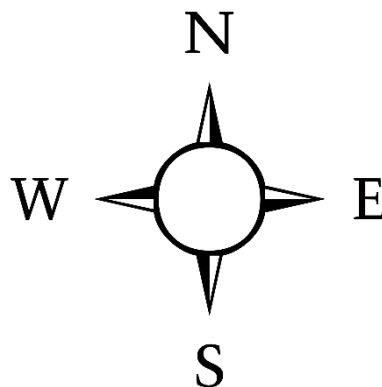
You will receive the initial direction the coders are heading in the form of a single letter.

(N)orth (E)ast (S)outh or (W)est.

You will then begin receiving numbers, representing degrees that you need to **rotate by**. **Every 45 degrees** of rotation means you need to go onto the **next direction**.

- To go from **North** to **East** you need **45 degrees**.
- To go from **North** to **West** you need **135 degrees**.
- To go from **South** to **North** you need **-90 degrees**.

Positive numbers indicate **clockwise** rotation, while **negative** numbers indicate **counter-clockwise** rotation. Until you receive the **"END"** command you should keep **rotating around** the compass. When you receive END on the console you should print the final direction.



For example if you receive **N** it means you start at **North**. After that you receive **45** as a number. This means you need to rotate **clockwise by 45 degrees** and end up looking **East**. If you then receive **-90** you should rotate **90 degrees counter-clockwise**.

The numbers to rotate by, will always be multiples of **45 - (45, 90, 135, 180, 360)**. After receiving END you should print **two lines**: the compass starting position and the end position you ended up in.

Input

The input will consist of two parts.

The first part will be a **single line** with **one character** in it representing the starting position.

- **Starting position** – a character that will always be one of the following 4: [N, E, S, W]

The second part will contain a **number of lines** containing a **number** representing the **degrees to rotate by**.

- **Degrees** – an **integer** indicating how many degrees to rotate by. This will always be an integer in the range [-360 ... 360]
- **END** – upon receiving "END" you should end your processing and print the final direction

The number of lines with degrees will never be more than 1000.

Output

You should output two lines.

First line will contain the starting position in the following format:

- "Starting Position: {startingPosition}"

Second line will contain the ending position after all rotations: in the format:

- "Position after rotating: {endingPosition}"

Example

Input	Output	Description
N 45 45 END	Starting Position: North Position After Rotating: South	We start at North. We first receive 45. Since it is positive we need to go clockwise on the compass. We rotate 45 degrees from North and end up on East. We then receive 45 again, so we rotate another 45 degrees from East and end up on South. We then receive END. So, we stop and print the result.
E 45 -90 180 -135 360 360 END	Starting Position: East Position After Rotating: East	Starting on East. East +45 -> South South -90 -> North North +180 -> North North -135 -> East (East + 360 -> East) x2

