Driving Home

TASK

You have just purchased a shiny new Segway human transporter to drive around town. Unfortunately, you are still learning how to drive it. You can steer it very well, but your control over speed is unpredictable.

When you have control of your Segway's control, your Segway takes exactly one minute to travel one block. When you lose control of your Segway's speed, however, your Segway takes you two blocks in a single minute.

You will always know at the beginning of a minute whether or not you will lose control of your Segway's speed. Also, at the start of any minute you can steer the Segway in any direction you choose. You cannot change direction except at the start of a minute. You are able to change direction at the start of each minute.

Write a program to help you get home. Your city is laid out as a perfect grid, with roads running north-south and east-west. At the start of each minute you find out whether or not you will have speed control for that minute. After learning the status of your speed control, determine which direction to steer your Segway. Minimize the number of minutes it takes you to get home.

Input: *standard input*

The first line of input contains two space-separated integers X and Y
(-100 <= X <= 100; -100 <= Y <= 100). X is the number of streets you are east of home (a negative X means you are west of home); Y is the number streets you are north of home (a negative Y means you are south of home).

For each minute, you will get a single line containing a single integer: 1 or 2. 1 means you have control of your Segway's speed and will go one block in a direction you choose. 2 means you do not have control of your Segway's speed and will go two blocks, also in any direction you choose.

Output: *standard output*

After learning the status of your speed, you must output a single character on a single line specifying the direction you want to direct your Segway. The character must be 'E' for east, 'W' for west, 'N' for north, or 'S' for south.

You must get home in 300 minutes or less to get any points. This will be possible for all input cases.

Your program must exit the first time you reach home.

Example exchange:

Input	Output	Location
3 -2		3 -2
1		
	W	2 -2
2		
	N	2 0
1		
	W	1 0
2		
	W	-1 0
1		
	E	0 0
	program exit	

CONSTRAINTS

Running time	1 second of CPU
Memory	64 MB

SCORING

Correctness: 20%

If you get home by the end of the 300th minute and properly exit, you will get at least 20% of the points for that test case. You must not output any invalid directions.

Optimality: 80%

Optimality of the program is measured by the number of minutes you take to get home on your Segway. Scoring will be based on the difference between your program's output and the optimal output.