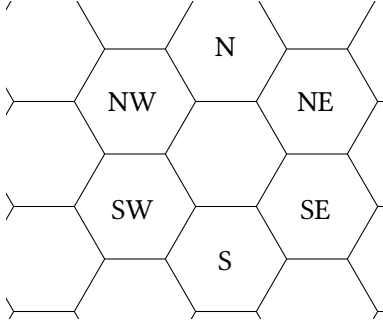


**Problem G**  
**Hex Ed**  
Time Limit: 1 Second

**Problem Description**

A program comes up to you, clearly in distress. “It’s my child process,” she says, “he’s gotten lost in an infinite grid!” Fortunately for her, you have plenty of experience with infinite grids. Unfortunately for you, it’s a hex grid.

The hexagons (“hexes”) in this grid are aligned such that adjacent hexes can be found to the north, northeast, southeast, south, southwest, and northwest:



You have the path the child process took. Starting where he started, you need to determine the fewest number of steps required to reach him. (A “step” means to move from the hex you are in to any adjacent hex.) For example:

- ne ne ne is 3 steps away.
- ne ne sw sw is 0 steps away (back where you started).
- ne ne s s is 2 steps away (se se).
- se sw se sw sw is 3 steps away (s s sw).

**Input File Format**

Input consists of a series of test cases. The first line of a test case will be an integer  $N$  representing the number of movements for the program. The following  $N$  lines will each specify a single direction (one of nw, n, ne, se, s, sw). The program should terminate when  $N = 0$ .

**Output Format**

Output the fewest number of steps (movement from one hex to another) it would take to reach the rogue program along the hexagonal grid.

### Sample Input

2  
ne  
s  
3  
ne  
s  
nw  
3  
n  
nw  
s  
0

### Output for the Sample Input

1  
0  
1