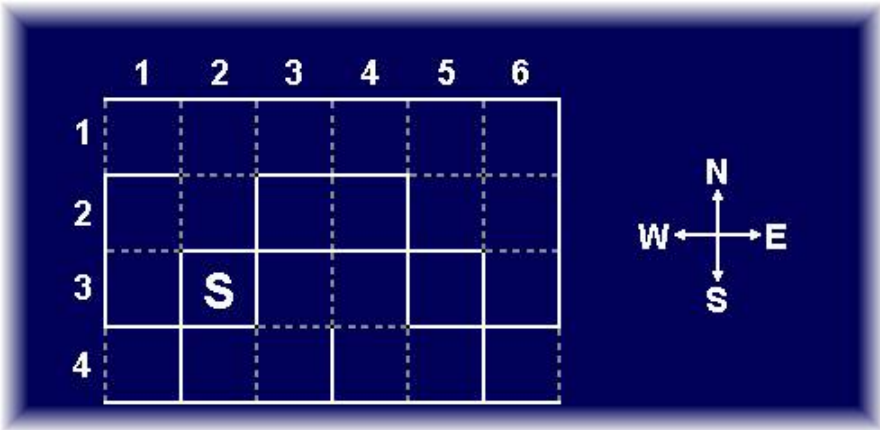


## 10384 The Wall Pushers

The figure below shows a maze with three exits. You are allowed to move between two squares within the maze if they are adjacent and 1) there is no wall separating the squares or 2) there is an inner wall between the squares, which may be pushed in the direction of movement. A wall can be pushed if there is no wall behind it. Notice that it's not allowed to push any wall that lies on the boundary of the maze.



Find the shortest path from the start position (S) to any of the exit

From the start position (S) in the figure above, it's possible to move north or east, but not west or south. If moving north, the wall between the squares (2, 3) and (2, 2) will be moved to the position between the squares (2, 2) and (2, 1). At this new position, it's not possible to move north again because there is a wall north of (2, 1).

Write a program that finds the shortest path from a given start position to any of the exits. You may assume there exists at least one solution for each maze.

### Input

The input file may contain several mazes to solve. Each maze description starts with a single line containing two integers  $x$  and  $y$  ( $1 \leq x \leq 6, 1 \leq y \leq 4$ ) which is the start position in the maze. Next follows four lines with six integers each. These integers  $p$  ( $0 \leq p \leq 15$ ) describe each square in the maze in the following way:  $p$  is the sum of 1 (if there is a wall west of the square), 2 (north), 4 (east) and 8 (south). Each inner wall will thus be mentioned twice. Each opening in the boundary is considered an exit.

The input ends with a maze with starting coordinates 0, 0 and should not be processed.

### Output

Output a single line for each maze with the description of a path with minimum length that leads to any of exits. Use the letters 'N', 'S', 'E' and 'W' to denote north, south, east and west, respectively. If there are several solutions with minimum length, display any one of them.

### Sample Input

```
2 3
10 2 10 10 2 6
```

```
3 12 11 14 9 4
13 15 3 6 15 13
14 11 12 9 14 11
0 0
```

**Sample Output**

```
NESESEENNWNWWWWW
```