

Problem 4—Sliding Number

The film *Time Code* reminds of the famous sliding number puzzle in which 15 tiles and one black space occupy a 4x4 grid. Tiles adjacent to the blank space horizontally or vertically can be slid into the blank space. The goal is to move the tiles around so that the numbers 1-15 are arranged numerically left to right and top to bottom with the blank space in the lower right corner. Not all initial arrangements result in a valid solution. In fact, exactly half of them do! Any arrangement may be moved to *either* the solution position *or* a “near solution” position in which the 14 and 15 tiles are reversed. Given an initial arrangement of tiles, you are to determine whether the puzzle can be solved from that position.

INPUT SPECIFICATION. You will be given a set of input cases, each of which will be a 16-byte string followed by <EOLN>. The last input case will be followed by an extra <EOLN>. Each input string will consist of 1-9, A-F (representing 10-15) and a space. Each of these characters will appear in the string exactly once, but they could appear in any order. This string represents a initial position in the 4x4 grid, with the first 4 bytes representing the first row, the second 4 bytes representing the second row, and so forth.

OUTPUT SPECIFICATION. The output cases should appear in the same order as the input cases. Each output case will be either “Case *c*: YES” or “Case *c*: NO” (where *c* is the case number) depending on whether the grid can be solved. <EOLN> follows each output case.

SAMPLE INPUT.

```
123456789ABCDEF • <EOLN>
123456789ABCDFE • <EOLN>
•123456789ABCDEF <EOLN>
<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case • 1 : • YES <EOLN>
Case • 2 : • NO <EOLN>
Case • 3 : • NO <EOLN>
<EOF>
```