

Q8: When Daylight Turns to Knight

Given a 4x4 chess board, place a knight on some square and mark that square number 1. The knight makes a number of moves, never visiting the same square twice, marking each visited square with the next larger number in sequence. A valid knight's move is shaped like an "L"; it moves either two squares vertically and one square horizontally, or two squares horizontally and one square vertically. The board shown here represents a **final valid** knight's path, starting in the lower-left corner (square marked 1) and ending in the upper right square (marked 15). Any square not visited by the knight has a mark of 0.

8	13	6	15
5	2	9	12
10	7	14	3
1	4	11	0

Your program must validate that a board containing a specific set of marked squares represents a **final valid** knight's path. A path is **valid** if the squares are marked such that moving from the square marked i to the square marked $i+1$ can be done using a valid knight's move. A path is **final** if the knight is unable to make any valid move from the highest marked square on the board.

Input

There will be four lines of input. Each line contains four integers separated by a single space. All integers will be greater than or equal to zero and less than or equal to 16. A zero represents a square that was never visited by the knight. No number (other than 0) will be repeated in the four lines of input.

Output

Your output will be a single string on a line by itself. Output TRUE (in capital letters) if the numbered board represents a **final valid** knight's path. Output FALSE (in capital letters) if this is not the case.

Sample Input and Output

Input	Output
0 5 10 15 13 2 7 4 6 9 14 11 1 12 3 8	TRUE
11 2 13 6 8 5 10 1 3 12 7 14 0 9 4 0	TRUE
1 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0	FALSE
1 3 14 15 7 12 9 6 4 2 16 13 11 8 5 10	FALSE