



BI-EP2: Efektivní programování 2

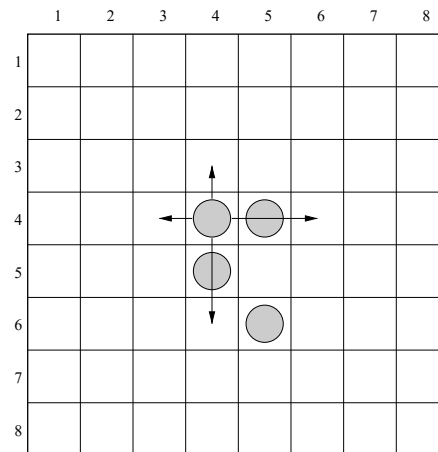
Solitaire

`solitaire.c`, `solitaire.cc`, `solitaire.java`

Solitaire is a game played on an 8×8 chessboard. The rows and columns of the chessboard are numbered from 1 to 8, from the top to the bottom and from left to right respectively.

There are four identical pieces on the board. In one move it is allowed to:

- move a piece to an empty neighboring field (up, down, left or right),
- jump over one neighboring piece to an empty field (up, down, left or right).



There are 4 moves allowed for each piece in the configuration shown above. As an example let's consider a piece placed in the row 4, column 4. It can be moved one row up, two rows down, one column left or two columns right.

Write a program that:

- reads two chessboard configurations from the standard input,
- verifies whether the second one is reachable from the first one in at most 8 moves,
- writes the result to the standard output.

Input Specification

The input contains several test cases. For each test case, each of two input lines contains 8 integers a_1, a_2, \dots, a_8 separated by single spaces and describes one configuration of pieces on the chessboard. Integers a_{2j-1} and a_{2j} ($1 \leq j \leq 4$) describe the position of one piece — the row number and the column number respectively.

Output Specification

For each test case, the output should contain one word “YES” if the configuration described in the second input line is reachable from the configuration described in the first input line in at most 8 moves, and one word “NO” otherwise.

Sample Input

```
4 4 4 5 5 4 6 5
2 4 3 3 3 6 4 6
4 4 4 5 5 4 6 5
2 4 3 3 3 6 2 2
```

Output for Sample Input

```
YES
NO
```