

# Panda Game | CodeChef

All submissions for this problem are available.

Dhoni got tired of playing cricket and thought of playing a board game involving pandas for relaxation. You are given a  $N \times N$  square grid where panda pieces are placed on the borders of the board. The borders consist of only the first row, the last row, the first column and the last column. In the following figure, the border cells are the ones which are marked with X. Each of the panda piece has a unique id which is a non negative 32 bit integer.

X	X	X	X	X
X				X
X				X
X				X
X	X	X	X	X

A panda piece can move in the following way. It can move  $N - 1$  positions either up, left, right or down and then take a 90 degree turn and moves one step further. This counts for a single move. But the move is possible only when the panda never goes out of the grid during the above mentioned process. Also, it can end up on a particular cell only if the cell is empty. Note that the movement of a panda is similar to a chess knight when  $N$  is 3.

Panda pieces are placed on all the boundary cells of the square grid except one grid which is left empty. Now, Dhoni is given an initial arrangement of the board, and he is required to find whether he can attain a particular arrangement or not.

## Input

- The first line of the input contains an integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.
- The first line of each test case contains a single integer  $N$  denoting the dimension of the board.
- The second line of the test case contains  $4 \times N - 4$  integers describing each boundary cells's panda id. ("-1" denotes the cell is empty and there is no panda piece placed at that particular cell position)
- The third line of the test case contains  $4 \times N - 4$  integers describing the final board configuration in the same manner.
- The order in which the boundary cells are described is the following. Starting from the top-most left cell, we describe the id of  $i^{\text{th}}$  piece of each cell situated at boundary in clockwise order ("-1" is given when that particular cell is empty).

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

For the above example, boundary cells are described in the order (0,0), (0,1), (0,2), (1,2), (2,2), (2,1), (2,0), (1,0).

## Output

For each test case, if it is possible to go from initial to final configuration, print “YES” (without quotes), else print “NO” (without quotes) in a separate line.

## Example

### Input:

```
1
3
1 2 3 4 7 6 5 -1
1 2 3 4 -1 6 5 7
```

### Output:

```
YES
```

## Explanation

The initial and final configurations of the grid are as follows

1	2	3
-1		4
5	6	7

1	2	3
7		4
5	6	-1

He moves the piece with id 7 to the empty square to achieve the required configuration.

Author:	<a href="#">codaitya</a>
Tags:	<a href="#">codaitya</a>
Date Added:	9-03-2016
Time Limit:	3 sec
Source Limit:	50000 Bytes
Languages:	C, CPP 4.3.2, CPP 4.9.2, CPP14, JAVA

## SUCCESSFUL SUBMISSIONS

1 of 1

User	Time	Mem	Lang	Solution
<a href="#">solvingmachine</a>	0.32	3.4M	C++14	<ul style="list-style-type: none"> <li><a href="#">View</a></li> </ul>