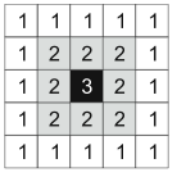
# C - Rotating Rings

**Input:** `standard input`
**Output:** `standard output`

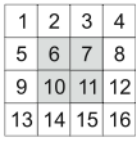
Any square grid can be viewed as one or more *rings*, one inside the other. For example, as shown in figure (a), a 5*5 grid is made of three rings, numbered 1, 2 and 3 (from outside to inside.)



| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 1 |
| 1 | 2 | 3 | 2 | 1 |
| 1 | 2 | 2 | 2 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Figure (a)

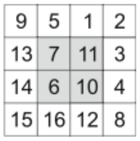
A square grid of size $N$ is said to be sorted, if it includes the values from 1 to $N^2$ in a row-majjor order, as shown in figure (b) for $N = 4$.



| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

Figure (b)

We would like to determine if a given square grid can be sorted *by only rotating its rings*. For example, the grid in figure (c) can be sorted by rotating the first ring two places counter-clockwise and rotating the second ring one place in the clockwise direction.



| | | | |
|---|---|---|---|
| 9 | 5 | 1 | 2 |
| 13 | 7 | 11 | 3 |
| 14 | 6 | 10 | 4 |
| 15 | 16 | 12 | 8 |

Figure (c)

## Input

Your program will be tested on one or more test cases. The first input line of a test case is an integer $N$, which is the size of the grid. $N$ input lines will follow, each line is made of $N$ integer values specifying the values in the grid in a row-major order. Note that $0 < N \leq 1000$ and grid values are natural numbers less than or equal to 1,000,000.

The end of the test cases is identified with a dummy test case with $N = 0$

## Output

For each test case, output the result on a single line using the following format:

`k.␣result`

Where $k$ is the test case number (starting at 1), ␣ is a single space, and `result` is `"YES"` or `"NO"` (without the quotes).

## Sample Input

```
4
9 5 1 2
13 7 11 3
14 6 10 4
15 16 12 8
3
1 2 3
5 6 7
8 9 4
0
```

## Sample Output

```
1. YES
2. NO
```