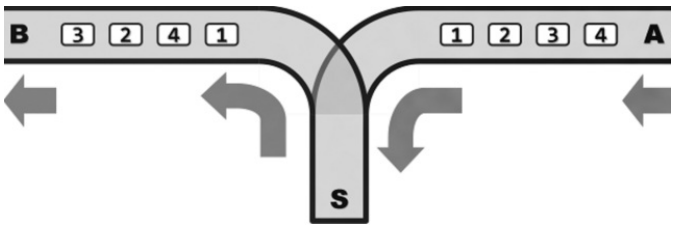
Train

This is a figure that shows the structure of a station for train dispatching.



In this station, A is the entrance for each train and B is the exit. S is the switching track. The coaches of a train can enter the switching track from direction A and must leave in direction B. Individual coaches can be disconnected from the rest of the train as they enter the switching track, so that they can be reorganized before they continue in direction B. If a coach enters the switching track from direction A, it must leave in direction B (i.e., it cannot return towards A). If a coach leaves in direction B, it cannot return back to the switching track.

Assume that a train consist of n coaches labeled {1, 2, …, n}. A dispatcher wants to know whether these coaches can pull out at B in the order of {a1, a2, …, an}.

Input

The 1st line contains an integer n (n <= 1,000) equal to the number of coaches, as described above. In each of the next lines of the input, except the last one, there is a permutation of 1, 2, . . . , n, this is the sequence {a1, a2, …, an} that the dispatcher would like to achieve as the coaches leave the switching track in direction B. The last line of the block contains just ‘0’ (to indicate the end of input).

Output

You should output the result for each permutation. If the sequence is feasible, output a "Yes", followed by a newline. If the sequence is infeasible, output a "No", followed by a newline.

Example 1 (pictured in the figure)

```
Input:
4
3 2 4 1
0

Output:
Yes
```

Example 2

```
Input:
5
1 2 3 4 5
4 5 1 3 2
0

Output:
Yes
No
```