C. Ravioli Sort

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Everybody knows of spaghetti sort. You decided to implement an analog sorting algorithm yourself, but as you survey your pantry you realize you're out of spaghetti! The only type of pasta you have is ravioli, but you are not going to let this stop you...

You come up with the following algorithm. For each number in the array a_i, build a stack of a_i ravioli. The image shows the stack for $a_i = 4$.

Arrange the stacks in one row in the order in which the corresponding numbers appear in the input array. Find the tallest one (if there are several stacks of maximal height, use the leftmost one). Remove it and add its height to the end of the output array. Shift the stacks in the row so that there is no gap between them. Repeat the procedure until all stacks have been removed.

At first you are very happy with your algorithm, but as you try it on more inputs you realize that it doesn't always produce the right sorted array. Turns out when two stacks of ravioli are next to each other (at any step of the process) and differ in height by two or more, the top ravioli of the taller stack slides down on top of the lower stack.

Given an input array, figure out whether the described algorithm will sort it correctly.

Input

The first line of input contains a single number n $(1 \le n \le 10)$ — the size of the array.

The second line of input contains n space-separated integers a_i $(1 \le a_i \le 100)$ — the elements of the array.

Output

Output "YES" if the array can be sorted using the described procedure and "NO" if it can not.

Examples

input
3 1 2 3
output
YES

input
3 3 1 2
output
NO

Note

In the second example the array will change even before the tallest stack is chosen for the first time: ravioli from stack of height 3 will slide on the stack of height 1, and the algorithm will output an array $\{2, 2, 2\}$.