

5963 Confusion in the Problem Set

A small confusion in a problem set may ruin the whole contest. So most of the problem setters try their best to remove any kind of ambiguity from the set. But sometimes it is not that important. For example the mock of last contest. As it was mock contest so we were not that serious with the set. We printed two problems, problem A in Page 1 and Problem B in Page 2. Then we remembered we have to give rule of the contest too. So we printed the rule page. But we did not notice that the rule page was printed with Page 2. We were stapling 3 pages together. First rule page, then Problem A and at the last Problem B. So page numbers were, 2, 1 and 2. This looked odd. But we already printed all the pages and if we want to fix the issue we had no other way but print all the three pages. One among us suggested an explanation — “Well, first 2 means there are 2 pages after this page. 1 also means there is 1 page after this page. But the last 2 means there are 2 pages before this page.” Interesting observation indeed. So we came up with a rule which is, page numberings of all the n pages are valid, if the page number at a page denotes number of pages before this page or number of page after this page.

So with this rule, $\{3, 1, 2, 0\}$ is valid but $\{3, 3, 1, 3\}$ is not valid.

Input

First line of the input file contains number of tests cases T (< 60).

Then T test cases follow. First line of each test case contains a positive number n ($\leq 10^4$) number of pages in the problem set. Then there are n space separated numbers in the following line each of which ranges from 0 to 10^6 .

Output

For each test case output the test case number. Then output ‘yes’ (without the quotes) if the pages can be shuffled somehow so that the pages numbering is valid. Otherwise output ‘no’.

For the exact format of the output please follow the sample.

Explanation of the samples:

1. The pages can be shuffled in several ways so that the page numbering is valid. One of the valid shuffles are 3, 1, 2, 0.
2. There is no valid way to shuffle these.

Sample Input

```
2
4
0 3 1 2
4
1 3 3 3
```

Sample Output

```
Case 1: yes
Case 2: no
```