PreContest Problem A : Doni and boring problems

Statement:

Doni has for long been a fan of competitive programming but has found that there are some problems that are too boring for her.
A problem is boring for her if the time it takes to solve strictly exceeds 60 minutes.
You are given the time in minutes it takes to solve a problem, determine whether Doni will find it boring.

Input :

The first line contains a single integer **T** (1≤T≤100) — the number of test cases. Then the test cases follow. Each test case consists of one line.

The first line contains an integer **n** (1≤ n ≤1000) — where n is the time in minutes it takes to solve a problem.

Output :

For each test case, output "YES" (in capital letters) without the quotes if Doni will find it boring and "NO" otherwise.

Example:

Input :

```
3
32
77
999
```

Output :

```
NO
YES
YES
```

PreContest Problem B : IMO Preparation

Statement:

Preparing for her first ever participation at the **International Mathematical Olympiad** (IMO), Sarah encountered a simple and tricky problem that she needs to prove.
In the process of proving it, she started by manually testing diffrent examples that she had in mind which turned out being a waste of time **especially with large numbers**. So as a software engineer student she decided to write a program that automatically gives her the result.
As one of her closest friends you are asked to help her writing this program.

The "IMO" Problem Statement :

Let $1 ≤ a0 < ... < an a finite sequence of positive integers.
Prove that there exists an integer k >= 1 such that :

$$a_k < \frac{a_0 + a_1 + \ldots + a_k}{k} \leq a_{k+1}$$

ak < (a0+a1+ ... + ak) / k ≤ a(k+1)

Input :

The first line contains a single integer **T** (1 ≤ T ≤ 100) — the number of test cases.

The first line of each test case contains an integer **N** (1 ≤ N ≤ 10^5) — where N is the length of the sequence.

The second line contains **N** integers **ai** (1 ≤ ai ≤ 10^4) describing the sequence.

Output :

For each test case, print one integer — the integer **k** mentionned in the **“IMO” Problem Statement** if **k** exists, if it doesn't then print **-1** .

Example:

Input :

```
2
4
1 2 9 10
6
4 5 6 7 8 9
```

Output :

```
1
3
```

In the first case k = 1, **ak** = 2

In the second case k = 3, **ak** = 7