All through history, some people have been interested in the solutions of polynomial equations. As everybody knows, in the Middle Ages wizards were all around. They claimed to be able to nd n solutions to any (univariate) polynomial equation of degree n. Of course, they sometimes needed to include some hocus-pocus like their magic number i, which they say is a solution to the equation $x^2 + 1 = 0$ (the second solution being $-i$).

But there are a few equations, for which most ordinary wizards failed to give n distinct solutions. Only the oldest and wisest wizards tried to be clever and bubbled something about multiplicity of roots — but nobody can possibly understand such excuses for nding fewer than n distinct roots.

Given a polynomial of degree n, nd out if wizards can possibly nd n distinct roots (including the magic ones using i), or if it is impossible - even for the wizards - to nd n distinct roots.

Input

Input starts with the number of test cases t ($1 \leq t \leq 100$) in a single line. Each test case consists of a single line that holds a series of integers (separated by single spaces). The rst integer is the degree n ($0 \leq n \leq 10$) of the polynomial in question. It is followed by the $n + 1$ coefficients $a_0 \dots a_n$ ($-30 \leq a_i \leq 30, a_0 = 0$) to form the equation $\sum_{i=0}^{n} a_i x^{n-i} = 0$.

Output

For each test case output '`Yes!`' on a single line (without the quotes) if the wizards have a chance (provided they are as good as they claim) to nd n distinct roots. Print '`No!`' on a single line (again without quotes) if there is no way any wizard can possibly nd n distinct roots.

Sample Input

```
5
2 1 1 1
2 1 2 1
4 1 2 1 2 1
4 1 2 2 2 1
4 1 0 2 0 1
```

Sample Output

```
Yes!
No!
Yes!
No!
No!
```