Task: SLO
Words

XVI OI, Stage III, Day one. Source file `slo.*` Available memory: 64 MB. *01.04.2009*

Let h be a function acting on strings composed of the digits `0` and `1`. The function h transforms the string w by replacing (independently and concurrently) every digit `0` with `1` and every digit `1` with the string „`10`". For example $h(„\texttt{1001}") = „\texttt{101110}"$, $h(„") = „"$ (i.e. h assigns an empty string to the empty string). Note that h is an injection, or a one-to-one function. By h^k we denote the function h composed with itself k times. In particular, h^0 is the identity function $h^0(w) = w$.

We are interested in the strings of the form $h^k(„\texttt{0}")$ for $k = 0, 1, 2, 3, \ldots$ This sequence begins with the following strings:

„`0`", „`1`", „`10`", „`101`", „`10110`", „`10110101`".

We call the string x a *substring* of the string y if it occurs in y as a contiguous (i.e. one-block) subsequence. A sequence of integers $k_1, k_2, \ldots, k_n$ is given. Your task is to check whether a string of the form

$$h^{k_1}(„\texttt{0}") \cdot h^{k_2}(„\texttt{0}") \cdots h^{k_n}(„\texttt{0}")$$

is a substring of $h^m(„\texttt{0}")$ for some m.

Input

The first line of the standard input contains a single integer t, $1 \leqslant t \leqslant 13$, denoting the number of test units. The first line of each test unit's description contains one integer n, $1 \leqslant n \leqslant 100\,000$. The second line of each description holds n non-negative integers $k_1, k_2, \ldots, k_n$, separated by single spaces. The sum of the numbers in the second line of any test unit description does not exceed 10 000 000.

Output

Your programme should print out t lines to the standard output, one for each test unit. Each line corresponding to a test unit should contain one word: `TAK` (*yes* in Polish — if $h^{k_1}(„\texttt{0}") \cdot h^{k_2}(„\texttt{0}") \cdots h^{k_n}(„\texttt{0}")$ is a substring of $h^m(„\texttt{0}")$ for some m in that test unit, or `NIE` (*no* in Polish) otherwise.

Example

For the input data:

```
2
2
1 2
2
2 0
```

the correct result is:

```
TAK
NIE
```

Explanation of the example: The string from the first test unit is „`110`" — it is a substring of $h^4(„\texttt{0}") = „\texttt{10110}"$ for example. In the second test unit there is a string „`100`", which is not a substring of $h^m(„\texttt{0}")$ for any m.