Strange Binary

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

A normal binary decomposition can be represented as $n = \overline{a_{31}a_{30}\cdots a_0}$, where $a_i \in \{0, 1\}$.

Now, there is a strange binary decomposition, still represented as $n = \overline{a_{31}a_{30}\cdots a_0}$, but it needs to satisfy:

1. For $i = 0, 1, \cdots, 31$, $a_i \in \{-1, 0, 1\}$.
2. For $i = 0, 1, \cdots, 30$, a_i and a_{i+1} cannot both be 0, i.e., $a_i^2 + a_{i+1}^2 \neq 0$.
3. $$\sum_{i=0}^{31} a_i 2^i = n$$

Given a non-negative integer n, find the above binary decomposition of n, or determine that it cannot be decomposed.

Input

The first line contains an integer T ($1 \leq T \leq 10^4$), representing the number of test cases.

For each test case, there is one line containing a non-negative integer n ($0 \leq n < 2^{30}$), representing the number to be decomposed.

Output

For each test case, output a string `YES` if n can be decomposed, or `NO` if n cannot be decomposed.

If the first line is `YES`, then output 32 integers $a_0, a_1, \ldots, a_{31}$, divided into 4 lines with 8 integers on each line.

Example

standard input	standard output
3 0 3 5	NO YES 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 YES -1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1