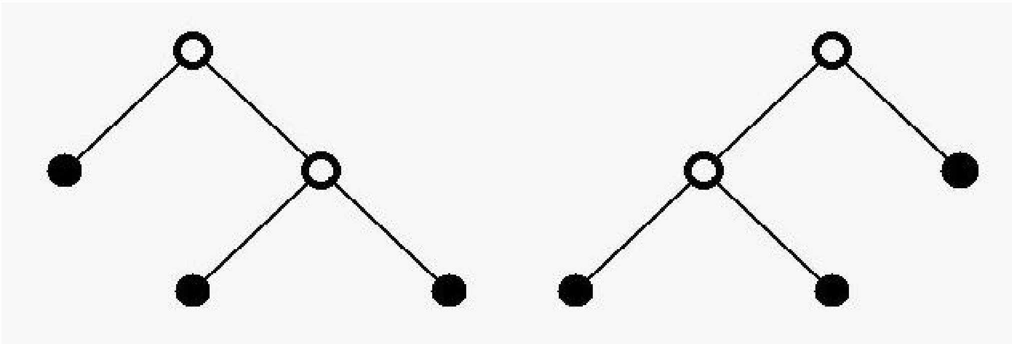
1134 Who's next?

Every computer science student knows binary trees. Here is one of many possible definitions of binary trees. Binary trees are defined inductively. A binary tree t is either an external node (leaf) '$\bullet$' or a single ordered pair (t_1, t_2) of two binary trees, left subtree t_1 and right subtree t_2 respectively, called an internal node '$\circ$'. Given an integer n, $B(n)$ is the set of trees with n leaves. For instance, the picture below shows the two trees of $B(3) = \{(\bullet, (\bullet, \bullet)), ((\bullet; , \bullet), \bullet)\}$.



Observe that those trees both have two internal nodes and a total of five nodes.

Given a tree t we define its unique integer identifier $N(t)$:

1. $N(\bullet) = 0$
2. $N(t_1, t_2) = 2^{n_1+n_2} + 2^{n_2}N(t_1) + N(t_2)$, where n_1 and n_2 are the number of nodes in t_1 and t_2 respectively.

For instance, we have $N(\bullet, \bullet) = 2^2 + 2^1 \times 0 + 0 = 4$, $N(\bullet, (\bullet, \bullet)) = 2^4 + 2^3 \times 0 + 4 = 20$ and $N((\bullet, \bullet), \bullet) = 2^4 + 2^1 \times 4 + 0 = 24$.

The ordering $\succeq$ is defined on binary trees as follows:

$$\begin{aligned} \bullet &\succeq t \\ (t_1, t_2) &\succeq (u_1, u_2), \text{ when } t_1 \succeq u_1 \text{ and } t_1 \neq u_1, \text{ or } t_1 = u_1 \text{ and } t_2 \succeq u_2 \end{aligned}$$

Hence for instance, $(\bullet, (\bullet, \bullet)) \succeq ((\bullet, \bullet), \bullet)$ holds, since we have $\bullet \succeq (\bullet, \bullet)$.

Using the ordering $\succeq$, $B(n)$ can be sorted. Then, given a tree t in $B(n)$, we define $S(t)$ as the tree that immediately follows t in the sorted presentation of $B(n)$, or as the smallest tree in $B(n)$, if t is maximal in $B(n)$. For instance, we have $S(\bullet, \bullet) = (\bullet, \bullet)$ and $S(\bullet, (\bullet, \bullet)) = ((\bullet, \bullet), \bullet)$. By composing the inverse of N,S and N we finally define a partial map on integers.

$$s(k) = N(S(N^{-1}(k)))$$

Write a program that computes $s(k)$.

Input

The first input line contains an integer K, with $K > 0$. It is followed by K lines, each specifying an integer k_i with $1 \leq i \leq K$ and $0 \leq k_i < 2^{31}$.

Output

The output should consist of K lines, the i-th output line being $s(k_i)$, or 'NO' if $s(k_i)$ does not exist.

Sample Input

```
5
4
0
20
5
432
```

Sample Output

```
4
0
24
NO
452
```