Humbertov Moralov in his student days, enrolled in the Systems Engineering program at "University of the Missing Hill" in *The Heaven's Branch Office* (Colombia, South America). He then attended a course of Assembly Language (in the first half of 1997).

The course was fantastic, with very interesting topics such as bit manipulation, right shifts, left shifts, rotations, masks and other bitwise operations (and, or, xor, not). And the best, in that course he worked interesting programming challenges. One of those programming challenges follows.

The chosen programming challenge is named "Rotations". By that time he used to work with unsigned integers of eight bits (a byte), and the challenge consisted of figuring out if a particular number n could generate all the eight numbers from 0 to 7 taking groups of consecutive bits of size 3.

For example, the number 226 has the binary representation `11100010` ($b_7b_6b_5b_4b_3b_2b_1b_0$), the eight sequences of consecutive three-bits that can be generated are the following:

- $b_2b_1b_0 = (010)_2 = 2$
- $b_3b_2b_1 = (001)_2 = 1$
- $b_4b_3b_2 = (000)_2 = 0$
- $b_5b_4b_3 = (100)_2 = 4$
- $b_6b_5b_4 = (110)_2 = 6$
- $b_7b_6b_5 = (111)_2 = 7$
- $b_0b_7b_6 = (011)_2 = 3$
- $b_1b_0b_7 = (101)_2 = 5$

20 years have passed. Since today computers are more powerful and faster, the professor *Humbertov Moralov* wants you to solve this programming challenge for unsigned integers of 32 bits (four bytes). You must validate if the number n generates all the 32 numbers from 0 to 31 with sequences of consecutive five-bits.

## Input

Input begins with an integer $t$ ($1 \leq t \leq 3{*}10^5$), the number of test cases, followed by $t$ lines, each line containing an integer $n$ ($0 \leq n \leq 4{*}10^9$).

## Output

For each test case, you should print a single line containing the word '`yes`' or '`no`' depending if the integer number $n$ produces or not all the numbers from 0 to 31 with sequences of consecutive five-bits.

## Sample Input

```
15
65535
65259922
81354525
112805325
122525196
192052550
225525450
299525510
318353525
344152934
502445252
522595252
1296752550
3999995011
4000000000
```

## Sample Output

```
no
no
yes
no
no
yes
yes
no
yes
no
yes
yes
no
yes
no
```