Burizon Fort

`fort.c`, `fort.cpp`, `Fort.java`, `fort.py`

On Mars there is a huge complex of forts. Martians living in these forts are fond of eating Burizons and get very grumpy if they run out of them. Therefore it's quite common task to transport Burizons among these forts when the supplies are running low somewhere.

The transportation of Burizons is not so simple because of all the space debris falling down on the planet. If some space debris hits the transportation vehicle, not only the vehicle gets destroyed, but even all the supplies transported. That's why Martians can transport their Burizon supplies only at very precisely timed moments.

Luckily, Martian scientists developed a way to predict practical times when no space debris will be endangering the transport vehicles. For this matter Martians developed a Martian UNIX timestamp, which is represented as a positive integer. Practical time for transportation is a Martian UNIX timestamp m, such that each positive integer smaller than m can be represented as a sum of distinct divisors of m.

As recognizing practical time for transport is not that easy for Martians, they will need you to help them to figure out which of the given timestamps are practical ones.

Input Specification

The first line contains one integer T ($1 \leq T \leq 100$), the number of timestamps to be checked. Each of the next T lines consists of one integer m ($1 \leq m \leq 10^{12}$), the Martian UNIX timestamp.

Output Specification

Output T lines, each with either "`Yes`" or "`No`", answering the question whether the i-th timestamp is practical.

Sample Input 1

```
5
4
6
10
12
15
```

Output for Sample Input 1

```
Yes
Yes
No
Yes
No
```