E. Tree Colorings

time limit per test: 4 seconds
memory limit per test: 256 megabytes

Consider a rooted undirected tree. Each vertex can be colored blue, green, or yellow. A coloring is called *beautiful* if it meets these conditions:

- the root of the tree is green;
- if you consider **all blue and green vertices**, they are reachable from each other without passing through any **yellow** vertices;
- if you consider **all yellow and green vertices**, they are reachable from each other without passing through any **blue** vertices;

You are given an integer m. Your task is to calculate the minimum number of vertices in a tree with **exactly** m *beautiful* colorings.

Input

The first line contains a single integer ($1 \le t \le 10^5$) — the number of test cases.

The only line of each test case contains a single integer m ($1 \le m \le 5 \cdot 10^5$).

Output

For each test case, print a single integer — the minimum number of vertices in a tree with **exactly** m *beautiful* colorings. If such a tree does not exist, print -1.

Example

input Copy

```
5
1
3
5
7
9
```

output Copy

```
1
2
3
4
3
```

Note

In the following notes, let g describe green color, b be blue, and y be yellow.

In the first example, consider a simple tree with just 1 vertex. This tree has exactly 1 beautiful coloring: the root is green.

In the second example, consider a simple tree with 2 vertices with a root at the 1-st vertex. There are exactly 3 beautiful colorings: $[g, g]$, $[g, b]$ and $[g, y]$.

In the third example, consider a bamboo tree with 3 vertices with a root at the 1-st vertex. There are exactly 5 beautiful colorings: $[g, g, g]$, $[g, g, b]$, $[g, g, y]$, $[g, b, b]$ and $[g, y, y]$.

In the fifth example, consider a tree with 3 vertices with a root at the 1-st vertex, and the other 2 vertices connected to it. There are exactly 9 beautiful colorings: $[g, g, g]$, $[g, g, b]$, $[g, g, y]$, $[g, b, g]$, $[g, b, b]$, $[g, b, y]$, $[g, y, g]$, $[g, y, b]$ and $[g, y, y]$.

→ Contest materials

- Announcement