6896 Eureka Theorem

A triangle number T_n ($T_n \geq 1$) is a figurate number that can be represented by a regular geometric arrangement of equally spaced points as illustrated in Figure 1.

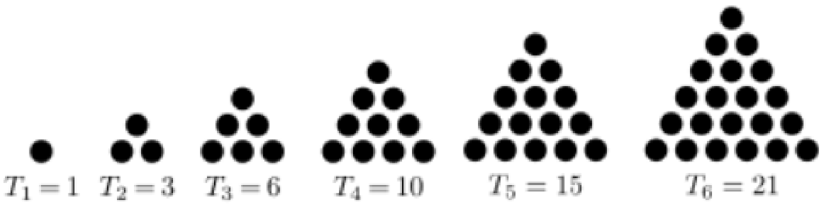

Figure 1.

The triangle number T_n for any positive integer $n \geq 1$ is given by the explicit formula:

$$T_n = 1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$$

In 1796, Gauss proved that every positive integer can be represented as a sum of *at most* three triangle numbers. For examples,

$$\begin{aligned} 4 &= T_1 + T_2 \\ 5 &= T_1 + T_1 + T_2 \\ 6 &= T_2 + T_2 \text{ or } 6 = T_3 \\ 10 &= T_1 + T_2 + T_3 \text{ or } 10 = T_4 \end{aligned}$$

This result is known as the Eureka theorem since he wrote in his diary "Eureka! $num = \Delta + \Delta + \Delta$" for commemorating the proof. We wonder if some positive integer can be represented as a sum of *exactly* three triangle numbers. As shown in the above examples, integers 5 and 10 can be represented as a sum of exactly three triangle numbers, but integers 4 and 6 cannot.

Given a positive integer, write a program to test whether or not the integer can be represented as a sum of exactly three triangle numbers that may not be distinct.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of a line containing a positive integer K ($3 \leq K \leq 1,000$).

Output

Your program is to write to standard output. Print exactly one line for each test case. Print '`1`' if the input number K can be represented as a sum of exactly three triangle numbers, and print '`0`' (zero), otherwise.

The following shows sample input and output for three test cases.

Sample Input

```
3
10
20
1000
```

Sample Output

```
1
0
1
```