Happy Numbers II

The process of "**breaking**" an integer is defined as summing the squares of its digits. For example, the
result of breaking the integer **125** is (**12** + **22** + **52**) = **30**. An integer **N** is **happy** if after "**breaking**" it
repeatedly the result reaches 1. If the result never reaches 1 no matter how many times the "**breaking**" is repeated, then N is not a happy number.

TASK

Write a program that given an integer **T** (number of test cases) and **T** integers, determines for each number whether it is a happy number or not.

CONSTRAINTS

1 ≤ T ≤ 1,080,000

2 ≤ N ≤ 2,147,483,647 (number for determining whether it is happy or not)

Input

- The first line contains an integer **T**.
- next 1...T lines contain an integer **N** for detemining whether it is happy or not.

Output

- T lines containing a single integer **N** which is the number of times the process had to be done to determine that N is happy, or **-1** if **N** is not happy.

Example

Input:
2
19
204

Output:
4
-1

1) 19 : $1^2 + 9^2 = 82$
2) 82 : $82 + 2^2 = 68$
3) 68 : $6^2 + 8^2 = 100$
4) 100 : $1^2 + 0^2 + 0^2 = 1$

The solution for 19 is 4 because we discovered that the integer 19 is happy after we repeated the process 4 times.

204 –> 20 –> 4 –> 16 –> 37 –> 58 –> 89 –> 145 –> 42 –> 20 –> 4 –> 16 –> 37 –> 58 –> 89 –> 145 ……..

204 is not a happy number because after breaking it several times the results start repeating so we can deduce that if we continue breaking it, the result will never reach 1.