Happy Numbers I

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 ($\mathbf{1^2} + \mathbf{2^2} + \mathbf{5^2}$) = **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 N, determines whether it is a happy number or not.

CONSTRAINTS

2 ≤ **N** ≤ 2,147,483,647

Input

A single line containing a single integer **N**.

Output

A single line containing a single integer **T** 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:
19

Output:
4

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

Input:
204

Output:
-1

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.

Number of input files is **32**.

Don't use pre-calculated values (Don't Cheat)!!!