

# Zeros in Fibonacci period

Perhaps the first thing one notices when the Fibonacci sequence is reduced mod  $p$  is that it seems periodic.

For example :

$F \pmod{2} = \underline{0} \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ \dots$

$F \pmod{3} = \underline{0} \ 1 \ 1 \ 2 \ \underline{0} \ 2 \ 2 \ 1 \ 0 \ 1 \ 1 \ 2 \ \dots$

$F \pmod{5} = \underline{0} \ 1 \ 1 \ 2 \ 3 \ \underline{0} \ 3 \ 3 \ 1 \ 4 \ \underline{0} \ 4 \ 4 \ 3 \ 2 \ \underline{0} \ 2 \ 2 \ 4 \ 1 \ 0 \ 1 \ 1 \ 2 \ 3 \ \dots$

We define  $Z(p)$  the number of zeros in fundamental period of Fibonacci numbers mod  $p$  (if it is periodic).

We just saw that  $Z(2) = 1$ ,  $Z(3) = 2$ , and  $Z(5) = 4$ .

## Input

The first line contains  $T$ , the number of test cases.

Each of the next  $T$  lines contains a prime number  $p$ .

## Output

For each test case, print  $Z(p)$ , or "Not periodic." without quotes if need.

## Example

**Input:**

3  
2  
3  
5

**Output:**

1  
2  
4

## Constraints

$1 < T < 10^5$

$1 < p < 10^{18}$ , a prime number

There's two input files, in the first one you are given all primes under  $10^6$ , in the second one lot of uniform randomly chosen primes.

**Warning** : Time limit had been changed, and could not allows slow languages to get AC here.

This problem allows easy coding using languages without bigNum library, but you need to be able to get at least some few points at [FIB64](#).

My best C-timing is 0.12s. (Edit : 0.03s with cluster change)

For other languages, take a look at [Z124H](#) with higher constraints. **Good luck, and have fun ;-)**