

Euclids algorithm revisited

Consider the famous Euclid algorithm to calculate the GCD of two integers (a, b):

```
int gcd(int a, int b) {
    while (b != 0) {
        int temp = a;
        a = b;
        b = temp % b;
    }
    return a;
}
```

for input (7, 3) the 'while' loop will run 2 times as follows: (7, 3) => (3, 1) => (1, 0)

Now given an integer N you have to find the smallest possible sum of two non-negative integers a, b ($a \geq b$) such that the while loop in the above mentioned function for (a, b) will run exactly N times.

Input

First line of input contains T ($1 \leq T \leq 50$) the number of test cases. Each of the following T lines contains an integer N ($0 \leq N \leq 10^{18}$).

Output

For each test case print the required answer modulo 1000000007 in a separate line.

Example

Input:

1
1

Output:

2

Explanation: (1,1) is the required pair.