Problem Statement:
Recently Manu visited the byteland. He was amazed with lots of different varieties of bytes he encountered there. Seeing this, he thought of a problem: Given an integer n, find out total possible bit string (either 0 or 1) of length n which don't have two contiguous zeroes in them. For example if n = 3, then total possible bit strings are 5 {010, 011, 101, 110, 111}. Now Manu started solving the problem but got busy with some important deployments. He asked you for the help. Please help him figure out the solution.

Input Format:
First line of test case contains an integer t denoting the number of test cases.
In next t lines, each line contains an integer n, denoting the length of bit string.

Output Format:
For t test cases, output the total number of bit string possible. Since this number can be very large, output it modulo 10^9+7.

Constraints:
1 <= t <= 10^3
1 <= n <= 10^15

Input Example:
2
2
3

Output Example:
3
5

Explanation:
For first test case (n = 2), the total possible bit strings are {01,10,11}. So answer is 3.
Second test case is same as provided in problem statement.

Subtask 1 (40 points):
t = 1
1 <= n <= 10^4

Subtask 2 (60 points):
original constraints