Fibonacci extraction Sum

Some people may found FIBOSUM a too easy problem. We propose here a useful variation.

$$\sum_{i=1}^{N} \mathrm{Fib}(ki + c)$$

Fib is the Fibonacci sequence:
For any positive integer i: if i<2 Fib(i) = i, else Fib(i) = Fib(i-1) + Fib(i-2)

Input

The first line of input contains an integer ***T***, the number of test cases.
On each of the next ***T*** lines, your are given tree integers ***c***, ***k***, ***N***.

Output

Print **Sum(Fib(*ki+c*) for *i* in [1..*N*])**.
As the answer could not fit in a 64-bit container, just output your answer modulo 1000000007.

Example

Input:
1
3 5 2

Output:
254

Explanations

Index-1 Fib sequence : 1, 1, 2, 3, 5, 8, 13, **21**, 34, 55, 89, 144, **233**, 377, 610, 987, ...
We want the 5*1+3 = 8th and 5*2+3 = 13th ones, thus the answer is 21 + 233 = 254.

Constraints

0 < T <= 60606
0 <= c < k <= 2^15
0 < N <= 10^18

The numbers ***c,k,N*** are uniform randomly chosen in their range.
For your information, constraints allow 1.3kB of Python3 code to get AC in 6.66s, it could be hard.
A fast C-code can get AC under 0.15s.
Warning: Here is Pyramid cluster, you can try the tutorial edition (clone with Cube cluster).
Have fun ;-)

Edit(2017-02-11) : With compiler changes, my fast C code ends in 0.01s, my Python3 ones in

0.31s. New TL is 0.5s.