```cpp
#include "../bits/stdc++.h"
// verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3613584#1

template <typename T>
using matrix = std::vector<std::vector<T>>;

constexpr long long mod = 1e9+7;

template <typename T>
matrix<T> modmul(const matrix<T> &l, const matrix<T> &r)
{
    matrix<T> ret(l.size(), std::vector<T>(r[0].size()));
    for (int i = 0; i < (int)l.size(); i++)
    {
        for (int k = 0; k < (int)r.size(); k++)
        {
            for (int j = 0; j < (int)r[0].size(); j++)
            {
                ret[i][j] += l[i][k] * r[k][j] % mod;
                ret[i][j] %= mod;
            }
        }
    }
    return ret;
}

template <typename T>
std::vector<T> modmul(const matrix<T> &a, const std::vector<T> &x)
{
    const int n = a.size();
    const int m = x.size();
    std::vector<T> ret(n);
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            ret[i] += a[i][j] * x[j] % mod;
            ret[i] %= mod;
        }
    }
    return ret;
}

template <typename T>
matrix<T> eye(const int n)
{
    matrix<T> ret(n, std::vector<T>(n));
    for (int i = 0; i < n; i++)
        ret[i][i] = 1;
    return ret;
}

template <typename T>
matrix<T> modpow(matrix<T> x, long long n)
{
    auto ret = eye<T>(x.size());
    while (n)
    {
        if (n & 1)
            ret = modmul(ret, x);
        x = modmul(x, x);
        n /= 2;
    }
    return ret;
}

```