

```

1  #include <bits/stdc++.h>
2  #define ll long long
3  #define MOD (int)(1e9+7)
4  using namespace std;
5
6  Matrix struct:
7  template<class T>
8
9  struct Matrix {
10     int n;
11     vector<vector<T>>> g;
12
13     Matrix(int n, function<T (int, int)> init = [](...){return 0;})
14         : n(n), g(n, vector<T>(n)) {
15         for(int i=0 ; i<n ; i++)
16             for(int j=0 ; j<n ; j++)
17                 g[i][j] = init(i, j);
18     };
19
20     auto& operator [] (int i) {
21         return g[i];
22     }
23
24     auto operator * (Matrix h) {
25         Matrix f(n);
26         for(int i=n; i--;)
27             for(int k=n; k--;)
28                 for(int j=n; j--;)
29                     f[i][j] = (f[i][j] + g[i][k] * h[k][j]) % MOD;
30         return f;
31     }
32
33     auto operator * (vector<T> h) {
34         vector<T> f(n, 0);
35         for(int i=n; i--;)
36             for(int j=n; j--;)
37                 f[i] = (f[i] + g[i][j] * h[j]) % MOD;
38         return f;
39     }
40
41     auto operator + (Matrix<T> h) {
42         Matrix<T> f(n);
43         for(int i=n; i--;)
44             for(int j=n; j--;)
45                 f[i][j] = (g[i][j] + h[i][j]) % MOD;
46         return f;
47     }
48
49     auto operator ^ (ll p) {
50         Matrix f(n, [](int i, int j) { return i==j; });
51         Matrix g = *this;
52         for( ; p ; p>>=1, g = g*g)
53             if(p&1) f = f * g;
54         return f;
55     }
56 };
57
58 // every row in the matrix product with the elements from vector
59 /* how to use:
60 Matrix<int> mat(10, [&](int i, int j) {
61     int x; cin>>x; return x;
62 }); */

```