

model matrix

```
1 struct Matrix{
2     #define type_data int
3     vector<vector<type_data>> v;
4     int row,col,mod;
5     Matrix(int r,int c,int mo){
6         row=r;
7         col=c;
8         mod=mo;
9         v.resize(r);
10        for(auto &a:v)
11            a.resize(c,0);
12    }
13    friend Matrix operator+(const Matrix& a,const Matrix& b){
14        assert(a.row==b.row&& a.col==b.col);
15        Matrix ret(a.row,b.col,a.mod);
16        for(int i=0;i<ret.row;i++)
17            for(int j=0;j<ret.col;j++)
18                ret.v[i][j]=(a.v[i][j]%ret.mod+b.v[i][j]%ret.mod)%ret.mod;
19        return ret;
20    }
21    friend Matrix operator-(const Matrix& a,const Matrix& b){
22        assert(a.row==b.row&& a.col==b.col);
23        Matrix ret(a.row,b.col,a.mod);
24        for(int i=0;i<ret.row;i++)
25            for(int j=0;j<ret.col;j++)
26                ret.v[i][j]=(a.v[i][j]%ret.mod-b.v[i][j]%ret.mod)%ret.mod;
27        return ret;
28    }
29    friend Matrix operator*(const Matrix& a,const Matrix& b){
30        assert(a.col!=b.col);
31        int len=a.col;
32        Matrix ret(a.row,b.col,a.mod);
33        for(int i=0;i<a.row;i++)
34            for(int j=0;j<b.col;j++)
35                for(int k=0;k<len;k++)
36                    ret.v[i][j]=(ret.v[i][j]+(a.v[i][k]%ret.mod)*(b.v[k]
37[j]%ret.mod)%ret.mod)%ret.mod;
38        return ret;
39    }
40    Matrix qpow(Matrix a,long long p){
41        assert(a.row==a.col);
42        Matrix ret(a.row,a.col,a.mod);
43        for(int i=0;i<ret.row;i++)
44            ret.v[i][i]=1;
45        for(;p>=1){
46            if(p&1)
47                ret=ret*a;
48            a=a*a;
49        }
50        return ret;
51    };
52 }
```

