

```

//递推公式黑科技
#include<bits/stdc++.h>
using namespace std;
///#define x first
///#define y second
#define PB push_back
///#define MP make_pair
///#define MEM(x,y) memset(x,y,sizeof(x));
///#define bug(x) cout<<"bug"<<x<<endl;
typedef long long ll;
//typedef pair<int,int> pii;
using namespace std;
//const int maxn=1e3+10;
const int mod=1000000007; //按实际改
ll powmod(ll a,ll b){ //快速幂
    ll res=1;a%=mod;
    assert(b>=0);
    while(b){
        if(b&1) res=res*a%mod; a=a*a%mod; b>>=1;
    }
    return res;
}
// head
namespace linear_seq {
    const int N=10010; //不需改
    ll res[N],base[N],_c[N],_md[N];
    vector<ll> Md;
    void mul(ll *a,ll *b,int k) {
        for(int i=0;i<k+k;i++) _c[i]=0;
        for(int i=0;i<k;i++) if (a[i])
            for(int j=0;j<k;j++) _c[i+j]=(_c[i+j]+a[i]*b[j])%mod;
        for (ll i=k+k-1;i>=k;i--) if (_c[i])
            for(int j=0;j<Md.size();j++)
                _c[i-k+Md[j]]=(_c[i-k+Md[j]]-
                _c[i]*_md[Md[j]])%mod;
        for(int i=0;i<k;i++) a[i]=_c[i];
    }
    int solve(ll n,vector<ll> a,vector<ll> b) {
        // a 系数 b 初值 b[n+1]=a[0]*b[n]+...
        //求出的是第n+1项
        ll ans=0,pnt=0;
        ll k=a.size();

```

```

assert(a.size()==b.size());
for(int i=0;i<k;i++) _md[k-1-i]=-a[i];_md[k]=1;
Md.clear();
for(int i=0;i<k;i++) if (_md[i]!=0) Md.push_back(i);
for(int i=0;i<k;i++) res[i]=base[i]=0;
res[0]=1;
while ((1ll<<pnt)<=n) pnt++;
for (ll p=pnt;p>=0;p--) {
    mul(res,res,k);
    if ((n>>p)&1) {
        for (ll i=k-1;i>=0;i--) res[i+1]=res[i];res[0]=0;
        for(int j=0;j<Md.size();j++) res[Md[j]]=
(res[Md[j]]-res[k]*_md[Md[j]])%mod;
    }
}
for(int i=0;i<k;i++) ans=(ans+res[i]*b[i])%mod;
if (ans<0) ans+=mod;
return ans;
}

vector<ll> BM(vector<ll> s) {
    vector<ll> C(1,1),B(1,1);
    int L=0,m=1,b=1;
    for(int n=0;n<s.size();n++) {
        ll d=0;
        for(int i=0;i<L+1;i++) d=(d+(ll)C[i]*s[n-i])%mod;
        if (d==0) ++m;
        else if (2*L<=n) {
            vector<ll> T=C;
            ll c=mod-d*powmod(b,mod-2)%mod;
            while (C.size()<B.size()+m) C.PB(0);
            for(int i=0;i<B.size();i++) C[i+m]=
(C[i+m]+c*B[i])%mod;
            L=n+1-L; B=T; b=d; m=1;
        } else {
            ll c=mod-d*powmod(b,mod-2)%mod;
            while (C.size()<B.size()+m) C.PB(0);
            for(int i=0;i<B.size();i++) C[i+m]=
(C[i+m]+c*B[i])%mod;
            ++m;
        }
    }
    return C;
}

int gao(vector<ll> a,ll n) {
    vector<ll> c=BM(a);

```

```

        c.erase(c.begin());
        for(int i=0;i<c.size();i++) c[i]=(mod-c[i])%mod;
        return solve(n,c,vector<ll>
(a.begin(),a.begin()+c.size()));
    }
};

//用的时候只用改mod的值和前几项的数值
int main(){
    ll n;
    while(~scanf ("%lld", &n)){ //求第n项 //一般放入前8项
        printf("%lld\n",linear_seq::gao(vector<ll>
{1,5,11,36,95,281,781,2245},n-1));
    }
}

```