高斯消元

```
#include<cstdio>
#include<iostream>
#include<algorithm>
#include<cmath>
using namespace std;
double map[111][111];
double ans[111];
double eps=1e-7;
int main(){
        freopen("testdata.in","r",stdin);
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n+1;j++)
            scanf("%lf",&map[i][j]);
    for(int i=1;i<=n;i++){
        int r=i;
        for(int j=i+1;j<=n;j++)
            if(fabs(map[r][i])<fabs(map[j][i]))
                r=j;//find_the_biggest_number_of_the_first_column（at present)
        if(fabs(map[r][i])<eps){
            printf("No Solution");
            return 0;
        }
        if(i!=r)swap(map[i],map[r]);//对换一行或一列,属于找最大当前系数的其中一步。（这样就可以只处理当前行的系数
        double div=map[i][i];
        for(int j=i;j<=n+1;j++)
            map[i][j]/=div;
        for(int j=i+1;j<=n;j++){
            div=map[j][i];
            for(int k=i;k<=n+1;k++)
                map[j][k]-=map[i][k]*div;
        }
    }
    ans[n]=map[n][n+1];
    for(int i=n-1;i>=1;i--){
        ans[i]=map[i][n+1];
        for(int j=i+1;j<=n;j++)
            ans[i]-=(map[i][j]*ans[j]);
    }//回带操作
    for(int i=1;i<=n;i++)
        printf("%.2lf\n",ans[i]);
}
```

矩阵乘法

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
const int MAXN = 105;
const long long MOD = 1e9 + 7;
long long k;
int n;
class F
{
public:
    long long a[MAXN][MAXN];
    F(bool tag)
    {
        memset(a, 0, sizeof(a));
        if(tag)
                for (register int i = 1; i <= n; ++i)
                {
                    a[i][i] = 1;
                }
    }
    void input()
    {
        for (register int i = 1; i <= n; ++i)
        {
            for (register int j = 1; j <= n; ++j)
            {
                cin >> a[i][j];
            }
        }
    }
    void output()
    {
        for (register int i = 1; i <= n; ++i)
        {
            for (register int j = 1; j <= n; ++j)
            {
                cout << a[i][j] << ' ';
            }
            putchar('\n');
        }
    }
} t(0);
F operator*(F a,F b)
{
//      cerr << "*" << endl;
    F res(0);
//    res.output();
    for (register int k = 1; k <= n; ++k)
    {
        for (register int i = 1; i <= n; ++i)
        {
            for (register int j = 1; j <= n; ++j)
            {
                res.a[i][j] = (res.a[i][j] + a.a[i][k] * b.a[k][j] % MOD) % MOD;
            }
        }
    }
    return res;
```

```
}
inline F power(F a, long long k)
{
    F res(1);
    for (; k; k >>= 1)
    {
        if (k & 1)
        {
//              cerr << "k & 1" << endl;
//              res.output();
//              a.output();
            res = (res * a);
//            res.output();
                }
//              cerr << "a * a" << endl;
//              a.output();
//              cerr << endl;
        a = a * a;
//        cerr << k << endl;
//        res.output();
//        cerr << endl;
//        a.output();
//        cerr << endl;
    }
    return res;
}
signed main()
{
#ifndef ONLINE_JUDGE
    freopen("t.in", "r", stdin);
#endif
    cin >> n >> k;
    t.input();
    F ans = power(t, k);
    ans.output();
}
```

剩余定理

```
#include<cstdio>
#define ll long long
ll n,a[16],m[16],Mi[16],mul=1,X;
inline int rd(){
    int io=0;char in=getchar();
    while(in<'0'||in>'9')in=getchar();
    while(in>='0'&&in<='9')io=(io<<3)+(io<<1)+(in^'0'),in=getchar();
    return io;
}
void exgcd(ll a,ll b,ll &x,ll &y){
    if(b==0){x=1;y=0;return ;}
    exgcd(b,a%b,x,y);
    int z=x;x=y,y=z-y*(a/b);
}
int main(){
    n=rd();
    for(int t=1;t<=n;++t){
        int M=rd();m[t]=M;
        mul*=M;
        a[t]=rd();
    }
    for(int t=1;t<=n;++t){
        Mi[t]=mul/m[t];
        ll x=0,y=0;
        exgcd(Mi[t],m[t],x,y);
        X+=a[t]*Mi[t]*(x<0?x+m[t]:x);
    }
    printf("%lld",X%mul);
    return 0;
}
```

快速乘by xiejiadong

```
inline LL mul(LL x,LL y,LL p)
{
    x%=p;y%=p;
    if (p<=1000000000) return x*y%p;
    if (p<=1000000000000LL) return (((x*(y>>20)%p)<<20)+x*(y&((1<<20)-1)))%p;
    LL d=(LL)floor(x*(long double)y/p+0.5);
    LL res=(x*y-d*p)%p;if (res<0) res+=p;
    return res;
}
```

阶乘逆元

```
LL invf[M],fac[M]={1};
void fac_inv_init(LL n,LL p)
{
    for (int i=1;i<n;i++)
        fac[i]=i*fac[i-1]%p;
    invf[n-1]=bin(fac[n-1],p-2,p);
    for (int i=n-2;i>=0;i--)
        invf[i]=invf[i+1]*(i+1)%p;
}
```

组合数

```
inline LL C(LL n,LL m){ // n >= m >= 0
    return n<m||m<0?0:fac[n]*invf[m]%MOD*invf[n-m]%MOD;
}
```

Lucas

```
LL Lucas(LL n, LL m) { // m >= n >= 0 mod is a prime
    return m ? C(n % MOD, m % MOD) * Lucas(n / MOD, m / MOD) % MOD : 1;
}
```

ex Lucas

```
const int N=1000000;
LL fac(const LL n,const LL p,const LL pk)
{
    if (!n) return 1;
    LL ans=1;
    for (int i=1;i<pk;i++)
        if (i%p) ans=ans*i%pk;
    ans=bin(ans,n/pk,pk);
    for (int i=1;i<=n%pk;i++)
        if (i%p) ans=ans*i%pk;
    return ans*fac(n/p,p,pk)%pk;
}
LL inv(const LL a,const LL p){LL x,y;ex_gcd(a,p,x,y);return (x%p+p)%p;}
LL C(const LL n,const LL m,const LL p,const LL pk)
{
    if (n<m) return 0;
    LL f1=fac(n,p,pk),f2=fac(m,p,pk),f3=fac(n-m,p,pk),cnt=0;
    for (LL i=n;i;i/=p) cnt+=i/p;
for (LL i=m;i;i/=p) cnt-=i/p;
    for (LL i=n-m;i;i/=p) cnt-=i/p;
    return f1*inv(f2,pk)%pk*inv(f3,pk)%pk*bin(p,cnt,pk)%pk;
}
LL a[N],c[N];
int cnt;
inline LL CRT()
{
    LL M=1,ans=0;
    for (int i=0;i<cnt;i++) M*=c[i];
    for (int i=0;i<cnt;i++) ans=(ans+a[i]*(M/c[i])%M*inv(M/c[i],c[i])%M)%M;
    return ans;
}
LL exlucas(const LL n,const LL m,LL p)
{//n>=m
    LL tmp=sqrt(p);cnt=0;
    for (int i=2;p>1&&i<=tmp;i++)
    {
        LL tmp=1;
        while (p%i==0) p/=i,tmp*=i;
        if (tmp>1) a[cnt]=C(n,m,i,tmp),c[cnt++]=tmp;
    }
    if (p>1) a[cnt]=C(n,m,p,p),c[cnt++]=p;
    return CRT();
}
```

Miller rubin

int 范围内只需检查 2, 7, 61 long long 范围 2, 325, 9375, 28178, 450775, 9780504, 1795265022 3E15内 2, 2570940, 880937, 610386380, 4130785767 4E13内 2, 2570940, 211991001, 3749873356

```
bool checkQ(LL a,LL n){
    if (n==2||a>=n) return 1;
    if (n==1||!(n&1)) return 0;
    LL d=n-1;
    while (!(d&1)) d>>=1;
    LL t=bin(a,d,n);  // 不一定需要快速乘
    while (d!=n-1&&t!=1&&t!=n-1){
        t=mul(t,t,n);
        d<<=1;
    }
    return t==n-1||d&1;
}

bool primeQ(LL n){
    int m=7,t[]={2,325,9375,28178,450775,9780504,1795265022};
    if (n<=1) return false;
    for (int i=0;i<m;i++) if (!checkQ(t[i],n)) return false;
    return true;
}
```