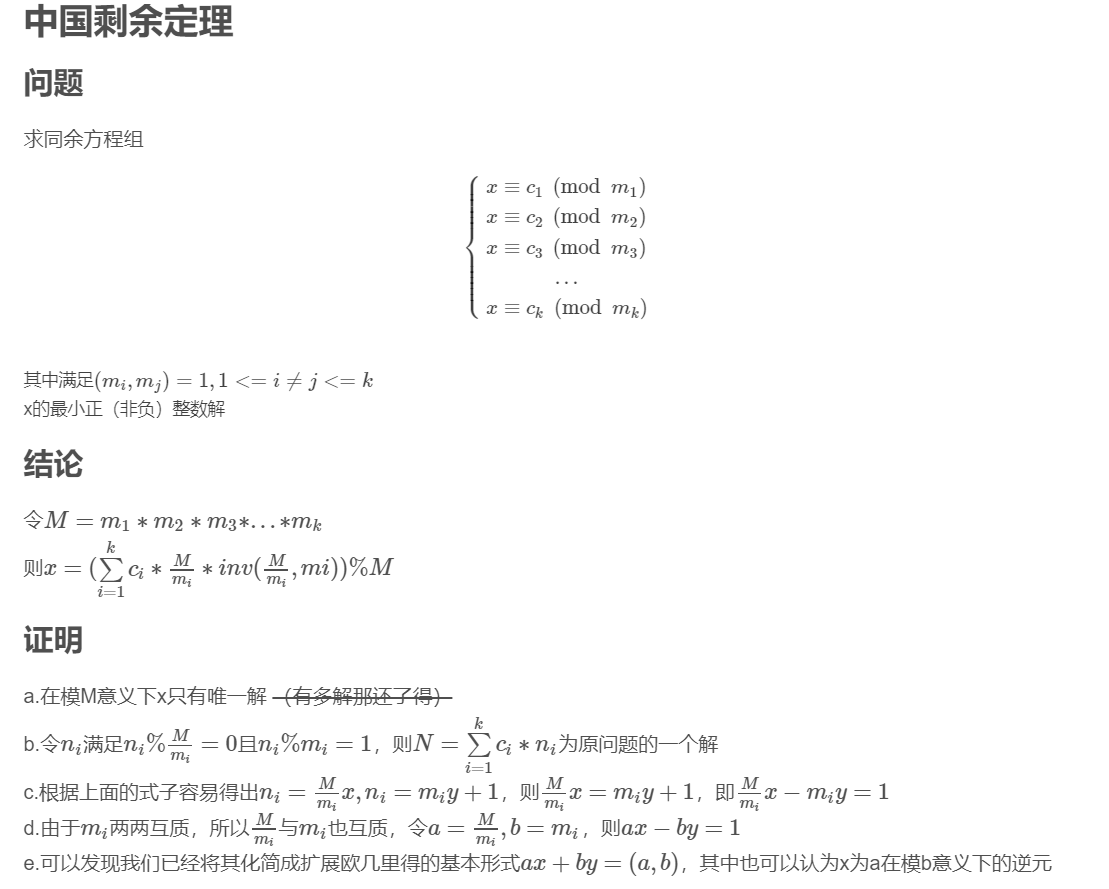
中国剩余定理 扩展中国剩余定理 Lucas 扩展Lucas



代码：codevs 3990

#include<algorithm>

#include<iostream>

#include<cstring>

#include<cstdio>

#include<cmath>

using namespace std;

#define LL long long

LL k,l,r,n,M,x,y,Min,ans,m[15],c[15];

void exgcd(LL a,LL b,LL &x,LL &y)

{

if (!b) x=1,y=0;

else exgcd(b,a%b,y,x),y-=a/b\*x;

}

int main()

{

scanf("%d%lld%lld",&k,&l,&r);

M=1LL;

for (int i=1;i<=k;++i)

{

scanf("%lld%lld",&m[i],&c[i]);

M\*=m[i];

}

for (int i=1;i<=k;++i)

{

LL a=M/m[i],b=m[i];

exgcd(a,b,x,y);

x=(x%b+b)%b;

if (!x) x+=b;

n+=c[i]\*a\*x;

}

n%=M;

if (!n) n+=M;

if (r>=n)

ans=(r-n)/M+1;

if (l>=n) ans=ans-((l-n)/M+1);

if ((l-n)%M==0) ++ans;

if (ans)

{

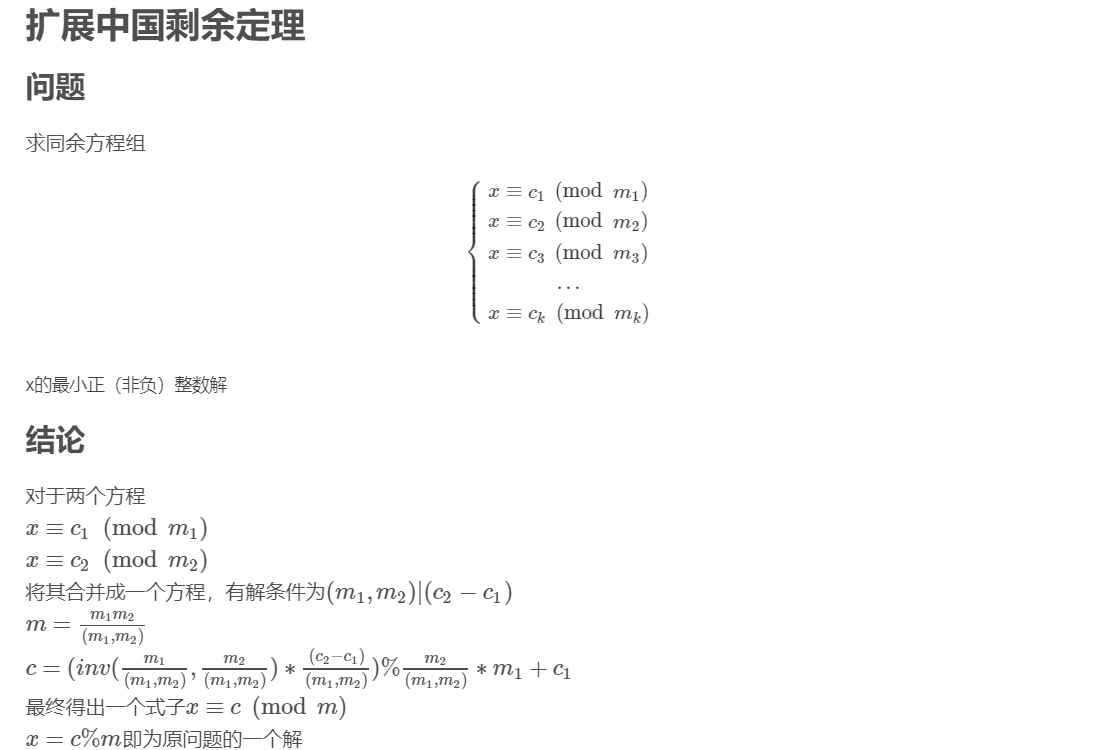
if (l<=n) Min=n;

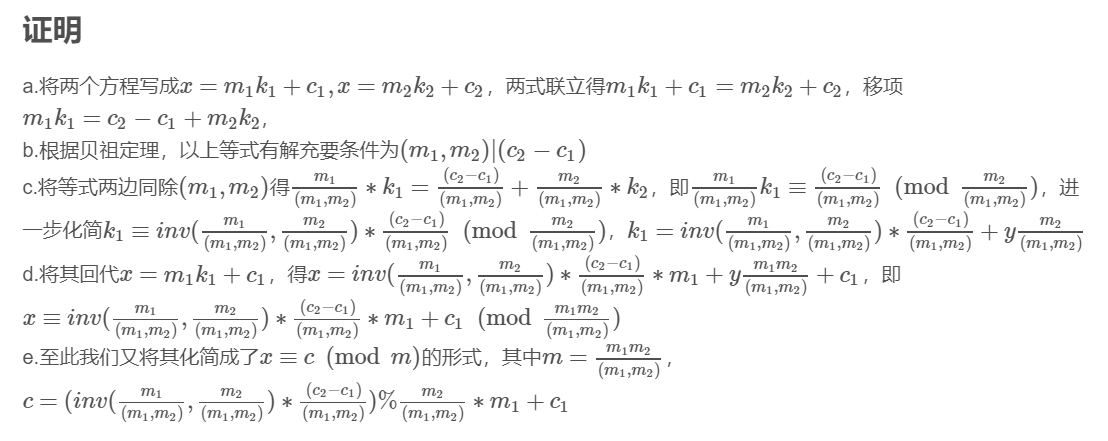
else Min=n+((l-n)/M+1)\*M;

}

printf("%lld\n%lld\n",ans,Min);

}





代码： pku 2891

#include<algorithm>

#include<iostream>

#include<cstring>

#include<cstdio>

#include<cmath>

using namespace std;

#define LL long long

#define N 1005

int k;

LL c[N],m[N],c1,c2,m1,m2,t;

bool flag;

LL gcd(LL a,LL b)

{

if (!b) return a;

else return gcd(b,a%b);

}

void exgcd(LL a,LL b,LL &x,LL &y)

{

if (!b) x=1LL,y=0LL;

else exgcd(b,a%b,y,x),y-=a/b\*x;

}

LL inv(LL a,LL b)

{

LL x=0LL,y=0LL;

exgcd(a,b,x,y);

x=(x%b+b)%b;

if (!x) x+=b;

return x;

}

int main()

{

while (~scanf("%d",&k))

{

flag=true;

for (int i=1;i<=k;++i)

scanf("%I64d%I64d",&m[i],&c[i]);

for (int i=2;i<=k;++i)

{

m1=m[i-1],m2=m[i],c1=c[i-1],c2=c[i];

t=gcd(m1,m2);

if ((c2-c1)%t!=0) {flag=false;break;}

m[i]=m1\*m2/t;

c[i]=inv(m1/t,m2/t)\*((c2-c1)/t)%(m2/t)\*m1+c1;

c[i]=(c[i]%m[i]+m[i])%m[i];

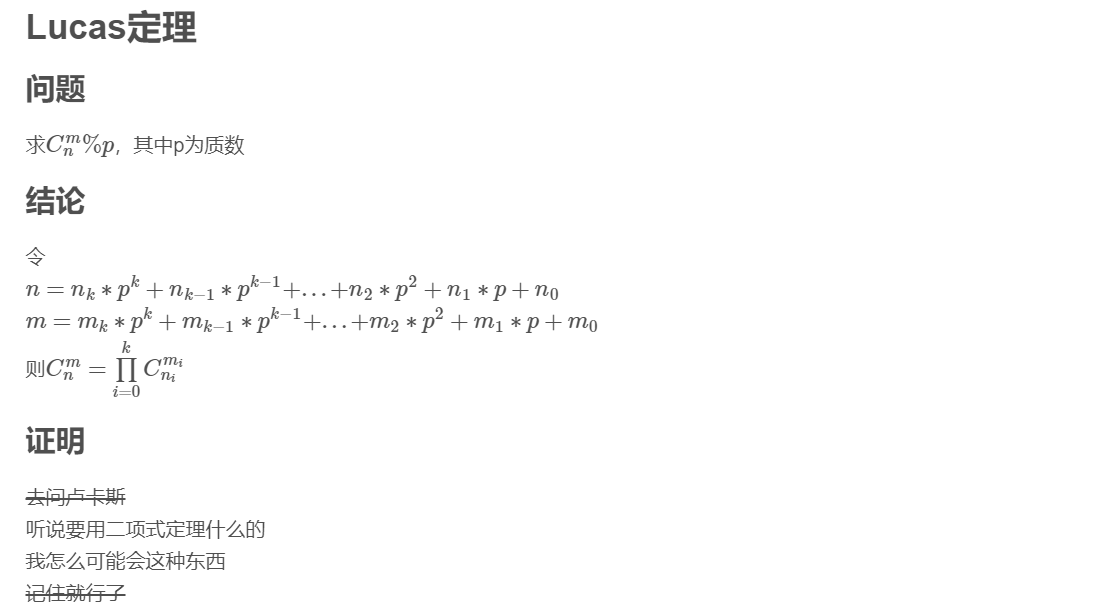
}

if (!flag) puts("-1");

else printf("%I64d\n",c[k]);

}

}



代码：zoj 3557

#include<algorithm>

#include<iostream>

#include<cstring>

#include<cstdio>

#include<cmath>

using namespace std;

#define LL long long

LL n,m,Mod;

LL fast\_pow(LL a,LL p)

{

LL ans=1LL;

for (;p;p>>=1,a=a\*a%Mod)

if (p&1)

ans=ans\*a%Mod;

return ans;

}

LL inv(LL x)

{

return fast\_pow(x,Mod-2);

}

LL C(LL n,LL m)

{

if (m>n) return 0LL;

LL up=1LL,down=1LL;

for (LL i=n-m+1;i<=n;++i) up=up\*i%Mod;

for (LL i=1;i<=m;++i) down=down\*i%Mod;

return up\*inv(down)%Mod;

}

LL lucas(LL n,LL m)

{

if (m>n) return 0LL;

LL ans=1;

for (;m;n/=Mod,m/=Mod)

ans=ans\*C(n%Mod,m%Mod)%Mod;

return ans;

}

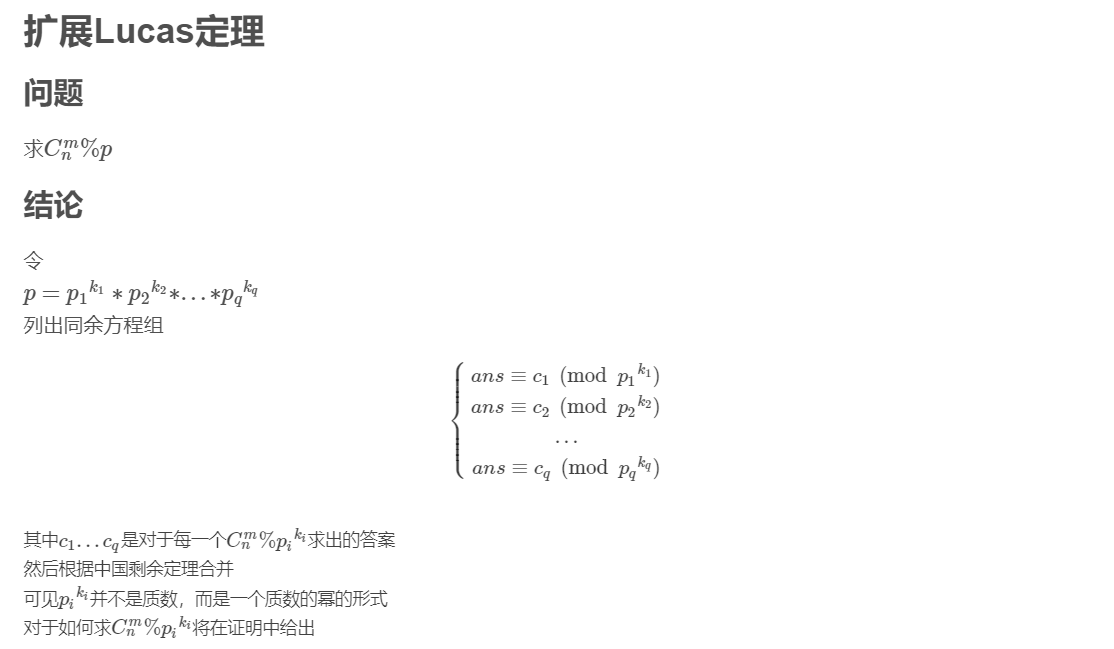
int main()

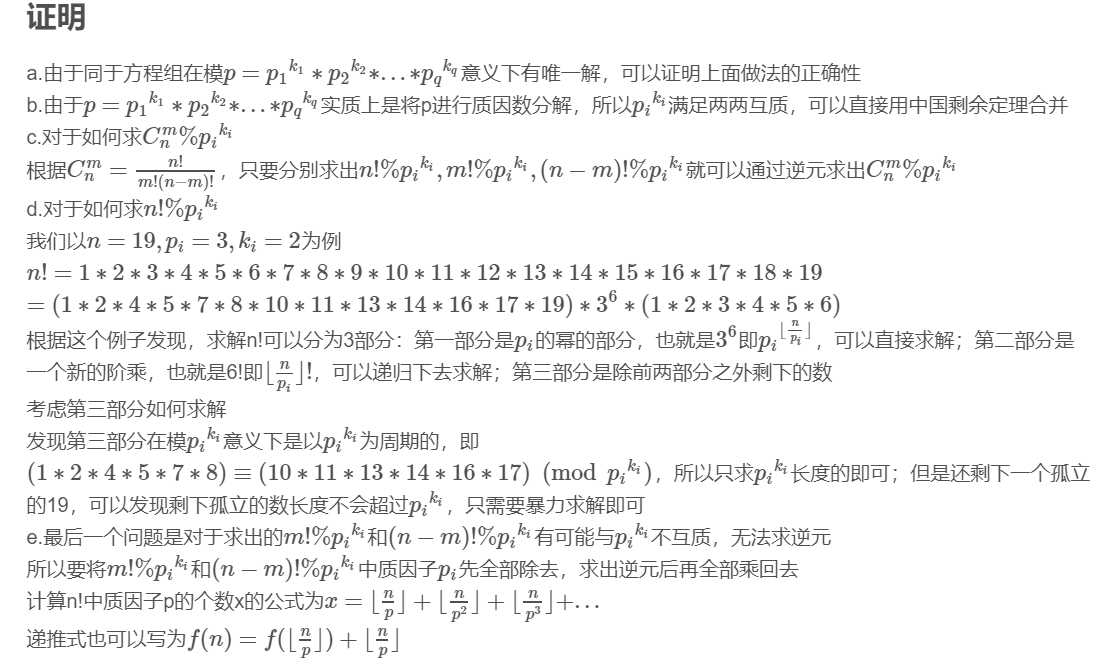
{

while (~scanf("%lld%lld%lld",&n,&m,&Mod))

printf("%lld\n",lucas(n-m+1,m));

}





代码：codeforces2015ICL,Finals,Div.1#J

#include<algorithm>

#include<iostream>

#include<cstring>

#include<cstdio>

#include<cmath>

using namespace std;

#define LL long long

LL n,m,MOD,ans;

LL fast\_pow(LL a,LL p,LL Mod)

{

LL ans=1LL;

for (;p;p>>=1,a=a\*a%Mod)

if (p&1)

ans=ans\*a%Mod;

return ans;

}

void exgcd(LL a,LL b,LL &x,LL &y)

{

if (!b) x=1LL,y=0LL;

else exgcd(b,a%b,y,x),y-=a/b\*x;

}

LL inv(LL A,LL Mod)

{

if (!A) return 0LL;

LL a=A,b=Mod,x=0LL,y=0LL;

exgcd(a,b,x,y);

x=((x%b)+b)%b;

if (!x) x+=b;

return x;

}

LL Mul(LL n,LL pi,LL pk)

{

if (!n) return 1LL;

LL ans=1LL;

if (n/pk)

{

for (LL i=2;i<=pk;++i)

if (i%pi) ans=ans\*i%pk;

ans=fast\_pow(ans,n/pk,pk);

}

for (LL i=2;i<=n%pk;++i)

if (i%pi) ans=ans\*i%pk;

return ans\*Mul(n/pi,pi,pk)%pk;

}

LL C(LL n,LL m,LL Mod,LL pi,LL pk)

{

if (m>n) return 0LL;

LL a=Mul(n,pi,pk),b=Mul(m,pi,pk),c=Mul(n-m,pi,pk);

LL k=0LL,ans;

for (LL i=n;i;i/=pi) k+=i/pi;

for (LL i=m;i;i/=pi) k-=i/pi;

for (LL i=n-m;i;i/=pi) k-=i/pi;

ans=a\*inv(b,pk)%pk\*inv(c,pk)%pk\*fast\_pow(pi,k,pk)%pk;

return ans\*(Mod/pk)%Mod\*inv(Mod/pk,pk)%Mod;

}

int main()

{

scanf("%I64d%I64d%I64d",&n,&m,&MOD);

for (LL x=MOD,i=2;i<=MOD;++i)

if (x%i==0)

{

LL pk=1LL;

while (x%i==0) pk\*=i,x/=i;

ans=(ans+C(n,m,MOD,i,pk))%MOD;

}

printf("%I64d\n",ans);

}