**//RSA**

#include<iostream>

using namespace std;

// Returns modulo inverse of a with respect to m using extended

int n=0;

int inv(int e, int r)

{

int d,b;

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

{

b = ((r\*i)+1);

if(b % e == 0)

break;

}

d = b/e;

return d;

}

int GCD(int x, int y)

{

while(x && y != 0)

{

if(x>y)

x=x%y;

else

y=y%x;

}

if(x==0)

return y;

else return x;

}

int modPow(int base, int exp, int n) {

base = base%n;

if (exp == 0)

return 1;

else if (exp == 1)

return base;

else if (exp%2 == 0)

return modPow(base\*base%n, exp/2, n);

else

return base\*modPow(base, exp-1, n)%n;

}

void RSA(int a, int b)

{

int r,e,d,M,C,result=0;;

n=a\*b;

r=(a-1)\*(b-1);

do

{

cout<<"\nEnter Encryption Key:";

cin>>e;

result=GCD(r,e);

if(result!=1)

{

cout<<"\nEncryption Key should be co-prime with "<<r<<"\n";

}

}while(result!=1);

d=inv(e,r);

cout<<"\nDecryption Key:"<<d<<"\n";

cout<<"\nEnter the number to encrypt:";

cin>>M;

C=modPow(M,e,n);

cout<<"\nCipher Text:"<<C<<"\n";

cout<<"\nEnter Cipher Text to decrypt:";

cin>>C;

M=modPow(C,d,n);

cout<<"\nOriginal No:"<<M<<"\n";

}

int main(void)

{

int p,q;

cout<<"Enter Prime No(1):";

cin>>p;

cout<<"Enter Prime No(2):";

cin>>q;

RSA(p,q);

return 0;

}