**20161113 이승수’s algorithm homework**

**<code>**

#include <stdio.h>

//#include <math.h>

#define MOD 100000000

typedef struct GCD{

int d;

int x;

int y;

}gcd;

FILE \*inF, \*outF;

gcd Extended\_Euclid(int a, int b);

void main()

{

inF = fopen("input.txt","r");

int p, q;

fscanf(inF,"%d %d",&p,&q);

printf("p,q=(%d,%d)",p,q);

fclose(inF);

int N = p\*q;

int PhiN = (p - 1)\*(q - 1);

int e;

for (e = 3; e < PhiN; e++)/\*Miller Rabin\*/

{

int relativePrime\_EPhiN = Euclid(PhiN, e);/\*Euclid\*/

if (relativePrime\_EPhiN == 1)

{

if (FermatTest(e, PhiN))

break;

}

}

printf("\npublic key(N,e):(%d,%d)+PhiN is %d",N,e,PhiN);

int d = 0;

for (d = 0; d < PhiN; d++)

{

if (e\*d%PhiN == 1)

break;

}

printf("\nprivate key(N,d): (%d,%d)",N,d);

int M = 65;//'A'

printf("\nPut Message: ");

scanf("%d",&M);

int C = pow(M, e) % N;

printf("\nM^e=%d",pow(M,e));

printf("\nMessage: %d -<encryption>--> encrypted C:%d", M, C);

printf(" -<decryption>--> Message: %d",pow(C,d)%N);

outF = fopen("output.txt","w");

fprintf(outF,"encrypted:%d,decrypted:%d",C,pow(C,d)%N);

fclose(outF);

}

int Fermat(int a,int p)

{

return (int)pow(a,p-1)%p;

}

int FermatTest(int p,int PhiN)

{

int fermatResult[100] = { 0 };

for (int k = 0; k < PhiN; k++)

{

fermatResult[k] = Fermat(k, p);

if (fermatResult[k] > 1)

return 0;

}

return 1;

}

int Euclid(int a, int b)

{

if (b == 0)

return a;

else

return Euclid(b,a%b);

}

gcd Extended\_Euclid(int a, int b)

{

gcd tmp;

if (b == 0)

{

tmp.d = a;

tmp.x = 1;

tmp.y = 0;

return tmp;

}

tmp = Extended\_Euclid(b, a % b);

gcd result;

result.d = tmp.d;

result.x = tmp.y;

result.y = (tmp.x - (a / b) \* tmp.y);

return result;

}

int pow(int a, int p)

{

int val = 1;

while (p > 0){

if (p % 2 == 1) val = (val \* a) % MOD;

p /= 2;

a = (a \*a) % MOD;

}

return val;

}