**CRT.h**

/\*

\* crt.h

\*

\* Created on: 11-Sep-2018

\* Author: dell

\*/

#ifndef CRT\_H\_

#define CRT\_H\_

class CRT {

public: int A[10] , N[10] , Y[10] , Z[310] , x , n;

public:

CRT();

int calculateX();

virtual ~CRT();

};

#endif /\* CRT\_H\_ \*/

**CRT.cpp**

**#include "crt.h"**

**#include** "EE.h"

**#include**<iostream>

**using** **namespace** std;

**CRT::CRT**() {

// **TODO** Auto-generated constructor stub

**for**(**int** i = 0 ; i < 3 ; i++)

{

A[i] = 0;

N[i] = 0;

Y[i] = 0;

Z[i] = 0;

}

x = 0;

n = 0 ;

}

**int** **CRT :: calculateX**()

{

EE e;

**int** muln = 1;

**for**(**int** i = 0 ; i < n ; i++)

{

muln = muln \* N[i];

}

cout << "\n\t multiplied n is: " << n << **endl**;

**for**(**int** i = 0 ; i < n ; i++)

{

Y[i] = muln/N[i];

cout << "Y[i] = " << Y[i] << **endl**;

}

**for**(**int** i = 0 ; i < n ; i++)

{

Z[i] = e.multInverse(N[i] , Y[i]);

cout << "Z[i] = " << Z[i] << **endl**;

}

**for**(**int** i = 0 ; i < n ; i++)

{

x = x + A[i]\*Y[i]\*Z[i];

}

x = x % muln;

**return** x;

}

**CRT::~CRT**() {

// **TODO** Auto-generated destructor stub

}

**MAIN.CPP**

/\*

\* main.cpp

\*

\* Created on: 11-Sep-2018

\* Author: dell

\*/

#include "crt.h"

#include<iostream>

using namespace std;

int main(void)

{

int x = 0;

CRT c;

cout << "\n\tEnter number of equations: ";

cin >> c.n;

cout << "for X = A (mod N)" << endl;

cout << "\n\tEnter the N values: " << endl;

for(int i = 0 ; i < c.n ; i++)

{

cin >> c.N[i];

}

cout << "\n\tEnter the A values: " << endl;

for(int i = 0 ; i < c.n ; i++)

{

cin >> c.A[i];

}

x = c.calculateX();

cout << "\n\t Value of X is: " << x;

return 0;

}

**EE.h**

**#ifndef EE\_H\_**

**#define** EE\_H\_

**class** EE {

**public**: **int** n1 , n2;

**int** q;

**int** r;

**int** t1 , t2;

**int** t;

**public**:

**EE**();

**int** **multInverse**(**int** n1 , **int** n2);

**virtual** **~EE**();

};

**#endif** /\* EE\_H\_ \*/

**EE.cpp**

**#include** "EE.h"

**#include**<iostream>

**using** **namespace** std;

**EE::EE**() {

// **TODO** Auto-generated constructor stub

n1 = 0;

n2 = 0;

t1 = 0;

t2 = 1;

t = 0;

q = 0;

r = 0;

}

**int** **EE :: multInverse**(**int** n1 , **int** n2)

{

t1 = 0;

t2 = 1;

**if**(n2 < 0)

{

n2 = n2 % n1;

n2 = n2 + n1;

//cout << "\n\t After mod n2 is: " << n2;

}

//cout << "\n\t q\tn1\tn2\tr\tt1\tt2\tt";

**while**(n1 != 1)

{

q = n1/n2;

r = n1%n2;

t = t1 - q\*t2;

//cout << "\n\t" << q << "\t" << n1 << "\t" << n2 << "\t" << r << "\t" << t1 << "\t" << t2 << "\t" << t;

n1 = n2;

n2 = r;

t1 = t2;

t2 = t;

**if**(r == 0)

{

**break**;

}

}

**if**(n1 == 1)

{

cout << "\n\tMultiplicative Inverse exists.";

**if**(t1 > 0)

{

cout << "\n\tMultiplicative inverse is: " << t1;

**return** t1;

}

**else**

{

t1 = t + t1;

cout << "\n\tMultiplicative inverse is: " << t1;

**return** t1;

}

}

**else**

{

//cout << "\n\tMultiplicative inverse does not exist";

**return** -1;

}

}

**EE::~EE**() {

// **TODO** Auto-generated destructor stub

}

OUTPUT

Enter number of equations: 4

for X = A (mod N)

Enter the N values:

2 3 5 7

Enter the A values:

1 1 3 1

multiplied n is: 4

Y[i] = 105

Y[i] = 70

Y[i] = 42

Y[i] = 30

Multiplicative Inverse exists.

Multiplicative inverse is: 1Z[i] = 1

Multiplicative Inverse exists.

Multiplicative inverse is: 1Z[i] = 1

Multiplicative Inverse exists.

Multiplicative inverse is: 3Z[i] = 3

Multiplicative Inverse exists.

Multiplicative inverse is: 4Z[i] = 4

Value of X is: 43