CNS LAB

Batch: B1

Assignment: 7

PRN No: 2020BTECS00006

Name: Samrat Vishwas Jadhav

Title of assignment: Implementation of Chinese Remainder Theorem

1. **Aim:**
   Implementation of Chinese Remainder Theorem

2. **Theory:**

Chinese Remainder Theorem:

If m1, m2, .., mk are pairwise relatively prime positive integers, and if a1, a2, .., ak are any integers, then the simultaneous congruences x ≡ a1 (mod m1), x ≡ a2 (mod m2), ..., x ≡ ak (mod mk) have a solution, and the solution is unique modulo m, where m = m1m2···mk .

Proof that a solution exists:

To keep the notation simpler, we will assume k = 4. Note the proof is constructive, i.e., it shows us how to actually construct a solution.

Our simultaneous congruences are

x ≡ a1 (mod m1), x ≡ a2 (mod m2), x ≡ a3 (mod m3), x ≡ a4 (mod m4).

Our goal is to find integers w1, w2, w3, w4 such that:

	value mod m_1	value mod m_2	value mod m_3	value mod m_4
w_1	1	0	0	0
w_2	0	1	0	0
w_3	0	0	1	0
w_4	0	0	0	1

Once we have found w1, w2, w3, w4, it is easy to construct x:

$x = a_1w_1 + a_2w_2 + a_3w_3 + a_4w_4$.

Moreover, as long as the moduli (m_1, m_2, m_3, m_4) remain the same, we can use the same w_1, w_2, w_3, w_4 with any a_1, a_2, a_3, a_4.

First define:

$z_1 = m / m_1 = m_2m_3m_4$

$z_2 = m / m_2 = m_1m_3m_4$

$z_3 = m / m_3 = m_1m_2m_4$

$z_4 = m / m_4 = m_1m_2m_3$

Note that

i) $z_1 \equiv 0 \pmod{m_j}$ for $j = 2, 3, 4$.

ii) ii) $\gcd(z_1, m_1) = 1$.
(If a prime p dividing m_1 also divides $z_1= m_2m_3m_4$, then p divides m_2, m_3, or m_4.)
and likewise for z_2, z_3, z_4.

Next define:

$y_1 \equiv z_1^{-1} \pmod{m_1}$

$y_2 \equiv z_2^{-1} \pmod{m_2}$

$y_3 \equiv z_3^{-1} \pmod{m_3}$

$y_4 \equiv z_4^{-1} \pmod{m_4}$

The inverses exist by (ii) above, and we can find them by Euclid's extended algorithm.

Note that

iii) $y_1z_1 \equiv 0 \pmod{m_j}$ for $j = 2, 3, 4$. (Recall $z_1 \equiv 0 \pmod{m_j}$)

iv) iv) $y_1z_1 \equiv 1 \pmod{m_1}$ and likewise for y_2z_2, y_3z_3, y_4z_4.

Lastly define:

$w_1 \equiv y_1z_1 \pmod{m}$

$w_2 \equiv y_2z_2 \pmod{m}$

$w_3 \equiv y_3z_3 \pmod{m}$

w4 ≡ y4z4 (mod m)

Then w1, w2, w3, and w4 have the properties in the above table.

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;
long long find_multiplicative_inverse(long long a, long long b)
{
    long long q, r, t1 = 0, t2 = 1, t, main_a = a;

    // cout << "______________________________________\n";
    // cout << "|\tQ\t|\tA\t|\tB\t|\tR\t|\tT1\t|\tT2\t|\tT\t|\n";
    // cout << "______________________________________\n";

    while (b > 0)
    {
        q = a / b;
        r = a % b;
        t = t1 - (t2 * q);
        // cout << "|\t" << q << "\t|\t" << a << "\t|\t" << b << "\t|\t" << r << "\t|\t" << t1 << "\t|\t" << t2 << "\t|\t" << t << "\t|\n";
        // cout << "______________________________________\n";

        a = b;
        b = r;
        t1 = t2;
        t2 = t;
    }

    // cout << "|\t" << q << "\t|\t" << a << "\t|\t" << b << "\t|\t" << r << "\t|\t" << t1 << "\t|\t" << t2 << "\t|\t" << t << "\t|\n";
    // cout << "______________________________________\n";

    if (t1 < 0)
```

```cpp
    {
        t1 += main_a;
    }
    return t1;
}
int main()
{

    cout << "____________________________\n";
    cout << "Lets Solve Chinese Remainder Theorem Problem  \n";
    cout << "____________________________\n";

    cout << "Suppose that equation needs to be in form of X = a (mod m)\n";

    cout << "How many equations you want to perfrom : \t";
    int count;
    cin >> count;
    cout << "\n____________________________\n";
    int M = 1;
    vector<int> a, m;
    for (int i = 0; i < count; i++)
    {
        cout << "Equation No : \t" << i + 1 << endl;
        cout << "Enter a :\t";
        int a_data;
        cin >> a_data;
        cout << "Enter m :\t";
        int m_data;
        cin >> m_data;
        a.push_back(a_data);
        m.push_back(m_data);
        cout << "\n____________________________\n";
        M = M * m_data;
    }
    cout << "\nValue of M  :\t" << M << endl;
    vector<long long> M_vector, M_inverse_vector;
```

```
    for (int i = 0; i < count; i++)
    {
        M_vector.push_back(M / m[i]); //caluculting M1,M2,M3
    }

    for (int i = 0; i < count; i++)
    {
        M_inverse_vector.push_back(find_multiplicative_inverse(m[
i], M_vector[i])); //m1,m2,m3 and M1,M2,M3---M1*M1^-1=1 mod m1;
    }

    long long sum = 0;
    for (int i = 0; i < count; i++)
    {
        sum += (a[i] * M_vector[i] * M_inverse_vector[i]);
    }
    long long ans = sum % M;
    cout << "\nAfter calculations :\n";
    cout << "______________________________________\n";
    cout << "|\tEq.
No\t|\ta[i]\t|\tm[i]\t|\tM[i]\t|\tM_inverse[i]\t|\n";
    cout << "______________________________________\n";

    for (int i = 0; i < count; i++)
    {
        cout << "|\t" << i + 1 << "\t|\t" << a[i] << "\t|\t" <<
m[i] << "\t|\t" << M_vector[i] << "\t|\t" << M_inverse_vector[i]
<< "\t|\n";
        cout << "______________________________________\n";
    }
    cout << "\nUsing formula X= E (a[i]*m[i]*m^-1[i]) mod M \n";
    cout << "Value of X is approximate equal to  :  " << ans;
    return 0;
}
```

Output:

```
PS D:\Final_BTech_Labs\CNS> cd "d:\Final_BTech_Labs\CNS\Assignment 7\" ; if ($?) { g++ crt.cpp -o crt } ; if ($?) { .\crt }
----------------------------
Lets Solve Chinese Remainder Theorem Problem
----------------------------
Suppose that equation needs to be in form of X = a (mod m)
How many equations you want to perfrom :        4


----------------------------
Equation No :   1
Enter a :       1
Enter m :       2


----------------------------
Equation No :   2
Enter a :       2
Enter m :       3


----------------------------
Equation No :   3
Enter a :       3
Enter m :       4


----------------------------
Equation No :   4
Enter a :       4
Enter m :       5


----------------------------

Value of M  :   120
```

```
After calculations :

---------------------------------------------
|       Eq. No  |       a[i]    |       m[i]    |       M[i]    |       M_inverse[i]    |
---------------------------------------------
|       1       |       1       |       2       |       60      |       0       |
---------------------------------------------
|       2       |       2       |       3       |       40      |       1       |
---------------------------------------------
|       3       |       3       |       4       |       30      |       1       |
---------------------------------------------
|       4       |       4       |       5       |       24      |       4       |
---------------------------------------------

Using formula X= E (a[i]*m[i]*m^-1[i]) mod M
Value of X is approximate equal to  :  74
```