**PRACTICAL - 06**

**Code:**

// Ashwin Navange A-38 CSE

#include <iostream>

using namespace std;

long long extended\_gcd(long long a, long long b, long long &x, long long &y) {

if (a == 0) {

x = 0;

y = 1;

return b;

}

long long x1, y1;

long long gcd = extended\_gcd(b % a, a, x1, y1);

x = y1 - (b / a) \* x1;

y = x1;

return gcd;

}

long long chinese\_remainder\_theorem(long long num[], long long rem[], int n) {

long long prod = 1;

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

prod \*= num[i];

}

long long result = 0;

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

long long pp = prod / num[i];

long long x, y;

extended\_gcd(num[i], pp, x, y);

result += rem[i] \* pp \* x;

}

return result % prod;

}

int main() {

int n;

cout << "Ashwin Navange A-38 CSE"<<endl;

cout << "Enter the number of congruences: ";

cin >> n;

long long num[n], rem[n];

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

cout << "Enter modulus (m" << i + 1 << "): ";

cin >> num[i];

cout << "Enter remainder (a" << i + 1 << "): ";

cin >> rem[i];

}

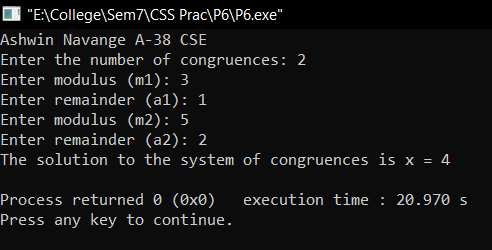
long long result = chinese\_remainder\_theorem(num, rem, n);

cout << "The solution to the system of congruences is x = " << result << endl;

return 0;

}

**Output:**

****