Lab 4

Name: Yuvraj Singh
Roll No: 23072021
M.Tech CSE

Problem: Given three positive integers A, B, and N, which represent a linear congruence of the form AX=B (mod N), the task is to print all possible values of X (mod N) i.e in the range [0, N-1] that satisfies this equation. If there is no solution, print -1.

Solution:
To find all possible values of $X \pmod N$ satisfying the linear congruence $AX \equiv B \pmod N$, where A, B, and N are integers. we can use the extended Euclidean algorithm. The algorithm will help to find the modular inverse of A modulo N, and then we can use it to find a particular solution. After that, you can use the period of the solution to generate all possible solutions.

Given the linear congruence:

$$AX \equiv B \pmod N$$

1. **Existence of Solution:** The linear congruence has a solution if and only if B is divisible by the greatest common divisor (gcd) of A and N. If $\gcd(A, N)$ does not divide B, then there is no solution.

Theorem 1. If $AX \equiv B \pmod N$ has a solution, then B is divisible by $\gcd(A, N)$.

Proof. Assume that $AX \equiv B \pmod N$ has a solution. This means there exists an integer X_0 such that $AX_0 \equiv B \pmod N$.
We can express this congruence as $AX_0 - B = kN$ for some integer k. Rearranging, we get $AX_0 - kN = B$.
Now, let $d = \gcd(A, N)$. Since d divides both A and N, it must divide AX_0 as well. Therefore, d must also divide $AX_0 - kN$.
This implies that d divides B, because $B = AX_0 - kN$.
Hence, we have shown that if $AX \equiv B \pmod N$ has a solution, then B is divisible by $\gcd(A, N)$. □

2. **Finding Modular Inverse:** If $\gcd(A, N)$ divides B, then a solution exists, and we can proceed to find the modular inverse of A modulo N. The modular inverse A^{-1} exists if A and N are relatively prime (i.e., $\gcd(A, N) = 1$).
Using the extended Euclidean algorithm, we find integers x and y such that $Ax + Ny = \gcd(A, N)$. If $\gcd(A, N) = 1$, then $Ax + Ny = 1$, and x is the modular inverse of A modulo N.

3. **Particular Solution:** Once we have the modular inverse A^{-1}, we can find a particular solution X_0 using the formula:

$$X_0 \equiv A^{-1} \cdot B \pmod N$$

4. **General Solution:** The general solution is then given by:

$$X \equiv (X_0 + k \cdot \frac{N}{\gcd(A, N)}) \pmod N$$

where k is an integer, and $\frac{N}{\gcd(A,N)}$ is the period of the solution. This expression ensures that all solutions are considered and lie in the range $[0, N-1]$.

Implementation in C++

```
#include <bits/stdc++.h>
using namespace std;

// calculate the greatest common divisor (GCD) using Euclid's Algorithm
long long 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 g = gcd(b % a, a, x1, y1);

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

    return g;
}

// find the modular inverse of 'a' modulo 'm'
long long inverse(long long a, long long m) {
    long long x, y;
    long long g = gcd(a, m, x, y);

    if (g != 1)  return -1; // Modular inverse doesn't exist
    else return (x % m + m) % m; // Ensure 'x' is positive
}

// solve the linear congruence AX     B (mod N)
void solve_linear_congruence(long long A, long long B, long long N) {
    long long A_inv = inverse(A, N);  // modular inverse of A modulo N

    if (A_inv == -1) {
        cout << -1 ; // No solution
        return;
    }

    // Find a particular solution X0 using the modular inverse
    long long X0 = (A_inv * B) % N;

    // Print all possible solutions in the range [0, N-1]
```

```cpp
    set<long long> X;
    for (long long k = 0; k < N; ++k)
        X.insert((X0 + k * (N / gcd(A, N))) % N);

    for (auto x: X) cout<<x<<" ";
    cout << endl;
}

int main() {
    long long A, B, N;
    cout << "Enter values for A, B, and N: ";
    cin >> A >> B >> N;

    solve_linear_congruence(A, B, N);

    return 0;
}
```

```
Enter values for A, B, and N: 3 2 7

Solutions: 3
```

```
Enter values for A, B, and N: 101 21 6564

Solutions: 1365
```

```
Enter values for A, B, and N: 5 7 10

Solutions: -1
```