Final Year B. Tech. (CSE) – I: 2022-23

4CS451: Cryptography and Network Security Lab

Assignment No. 9

PRN: 2019BTECS00077 **Batch: B7**

Full name: Biradar Avinash Vishnu

Title: Chinese Remainder Theorem

Objective: To find value of number x which divided by some co-prime numbers gives particular remainders.

Code:

```
      // Chinese remainder

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

// multiplicative-inverse-under-modulo-m/
int inv(int a, int m)
{
    int m0 = m, t, q;
    int x0 = 0, x1 = 1;

    if (m == 1)
        return 0;

    // Apply extended Euclid Algorithm
    while (a > 1) {
        // q is quotient
        q = a / m;

        t = m;

        // m is remainder now, process same as
        // euclid's algo
        m = a % m, a = t;
```

```
        t = x0;

        x0 = x1 - q * x0;

        x1 = t;
    }

    // Make x1 positive
    if (x1 < 0)
        x1 += m0;

    return x1;
}

int gcd(int a,int b){

    if(!b)
     return a;
    return gcd(b,a%b);
}

// k is size of num[] and rem[]. Returns the smallest
// number x such that:
// x % num[0] = rem[0],
// x % num[1] = rem[1],
// ..................
// x % num[k-2] = rem[k-1]
int findMinX(vector<int> &num,vector<int> &rem, int k)
{
    // Compute product of all numbers
    int prod = 1;
    for (int i = 0; i < k; i++)
        prod *= num[i];

    cout<<"\n\nvalue of M is : "<<prod<<endl;

    // Initialize result
    int result = 0;

    // Apply above formula
    for (int i = 0; i < k; i++) {
        int pp = prod / num[i];
        cout<<"value of Mi of "<<num[i]<<" is : "<<pp<<endl;
        int inverse=inv(pp, num[i]);
        cout<<"value of inverse of "<<pp<<" is : "<<inverse<<endl;
```

```
        result += rem[i] * inverse * pp;
    }
    return result % prod;
}

// Driver method
int main(void)
{
    int n,val;
    cin>>n;
    vector<int> num,rem;

    for(int i=0;i<n;i++){
      cin>>val;
      num.push_back(val);
    }


    for(int i=0;i<n;i++){
      cin>>val;
      rem.push_back(val);
    }

    // first we need to check pairwise whether the no's are co-prime or
not;
    for(int i=0;i<n;i++){

        for(int j=i+1;j<n;j++){
           if(gcd(num[i],num[j])!=1){
             cout<<num[i]<<" "<<num[j]<<" are not co-prime hence can't
proceed\n";
             return 0;
           }
        }
    }
    cout<<"All pairs are co-prime can be move further...";
    int ans= findMinX(num, rem,n);;
    cout <<"\nx is " <<ans;
    return 0;
}
```

Output:

```
3
3 4 5
2 3 1
All pairs are co-prime can be move further...

value of M is : 60
value of Mi of 3 is : 20
value of inverse of 20 is : 2
value of Mi of 4 is : 15
value of inverse of 15 is : 3
value of Mi of 5 is : 12
value of inverse of 12 is : 3

x is 11
```

```
3
3 5 7
2 3 2
All pairs are co-prime can be move further...

value of M is : 105
value of Mi of 3 is : 35
value of inverse of 35 is : 2
value of Mi of 5 is : 21
value of inverse of 21 is : 1
value of Mi of 7 is : 15
value of inverse of 15 is : 1

x is 23
```