Name: Shreeyash S. Dongarkar

PRN: 22510025

BTech Final Year CSE

Cryptography and Network Security Lab (B – 1)

Assignment 3: Implementation of Euclidean and Extended Euclidean Algorithm

**Objectives**: To understand and implement the Chinese Remainder Theorem (CRT) for solving systems of simultaneous congruences with pairwise coprime moduli.

1. import java.util.Scanner;

2.

3. public class ChineseRemainderTheorem {

4.

5.     public static int[] extendedEuclidean(int a, int b) {

6.         if (b == 0) {

7.             return new int[]{a, 1, 0};

8.         }

9.         int[] vals = extendedEuclidean(b, a % b);

10.         int gcd = vals[0];

11.         int x1 = vals[2];

12.         int y1 = vals[1] - (a / b) \* vals[2];

13.         return new int[]{gcd, x1, y1};

14.     }

15.

16.     public static int modInverse(int a, int m) {

17.         int[] vals = extendedEuclidean(a, m);

18.         int gcd = vals[0];

19.         int x = vals[1];

20.         if (gcd != 1) {

21.             throw new ArithmeticException("Modular inverse does not exist for " + a + " mod " + m);

22.         } else {

23.             return (x % m + m) % m;

24.         }

25.     }

26.

27.     public static int chineseRemainder(int[] a, int[] n) {

28.         int k = a.length;

29.         int N = 1;

30.

31.         for (int ni : n) {

32.             N \*= ni;

33.         }

34.

35.         int result = 0;

36.

37.         for (int i = 0; i < k; i++) {

38.             int Ni = N / n[i];

39.             int Mi = modInverse(Ni, n[i]);

40.             result += a[i] \* Ni \* Mi;

41.         }

42.

43.         return ((result % N) + N) % N;

44.     }

45.

46.     public static void main(String[] args) {

47.         Scanner sc = new Scanner(System.in);

48.

49.         System.out.print("Enter number of congruences: ");

50.         int k = sc.nextInt();

51.

52.         int[] a = new int[k];

53.         int[] n = new int[k];

54.

55.         System.out.println("Enter congruences in the form: x ≡ ai (mod ni)");

56.

57.         for (int i = 0; i < k; i++) {

58.             System.out.print("Enter ai: ");

59.             a[i] = sc.nextInt();

60.             System.out.print("Enter ni: ");

61.             n[i] = sc.nextInt();

62.         }

63.

64.         try {

65.             int x = chineseRemainder(a, n);

66.             System.out.println("The smallest non-negative solution is: " + x);

67.         } catch (Exception e) {

68.             System.out.println("Error: " + e.getMessage());

69.         }

70.

71.         sc.close();

72.     }

73. }

