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

**Objective**: Develop a program to implement the Euclidean Algorithm and the Extended Euclidean Algorithm for computing the Greatest Common Divisor (GCD) of two integers and, in the extended version, also compute the modular inverse and the coefficients of Bézout's identity.

1. import java.util.Scanner;

2.

3. public class EuclideanAlgorithms {

4.

5.     public static int gcd(int a, int b) {

6.         while (b != 0) {

7.             int temp = b;

8.             b = a % b;

9.             a = temp;

10.         }

11.         return Math.abs(a);

12.     }

13.

14.     public static int[] extendedGcd(int a, int b) {

15.         if (b == 0) {

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

17.         }

18.         int[] vals = extendedGcd(b, a % b);

19.         int gcd = vals[0];

20.         int x1 = vals[1];

21.         int y1 = vals[2];

22.

23.         int x = y1;

24.         int y = x1 - (a / b) \* y1;

25.

26.         return new int[] { gcd, x, y };

27.     }

28.

29.     public static Integer modInverse(int a, int m) {

30.         int[] vals = extendedGcd(a, m);

31.         int gcd = vals[0];

32.         int x = vals[1];

33.

34.         if (gcd != 1) {

35.             return null;

36.         } else {

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

38.         }

39.     }

40.

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

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

43.

44.         System.out.print("Enter first integer (a): ");

45.         int a = sc.nextInt();

46.         System.out.print("Enter second integer (b): ");

47.         int b = sc.nextInt();

48.

49.         int gcdVal = gcd(a, b);

50.         System.out.println("\nUsing Euclidean Algorithm:");

51.         System.out.println("GCD(" + a + ", " + b + ") = " + gcdVal);

52.

53.         int[] result = extendedGcd(a, b);

54.         System.out.println("\nUsing Extended Euclidean Algorithm:");

55.         System.out.println("GCD = " + result[0]);

56.         System.out.println("Coefficients (x, y): x = " + result[1] + ", y = " + result[2]);

57.         System.out.println("Verification: " + a + "\*" + result[1] + " + " + b + "\*" + result[2] + " = " + result[0]);

58.

59.         if (result[0] == 1) {

60.             Integer inv = modInverse(a, b);

61.             System.out.println("\nModular Inverse of " + a + " modulo " + b + " = " + inv);

62.         } else {

63.             System.out.println("\nModular inverse does not exist since GCD ≠ 1.");

64.         }

65.

66.         sc.close();

67.     }

68. }



