**Cryptography and Network Security**

**Lab**

**Assignment No. 3**

**Name : Sourabh Shankar Patil**

**PRN : 21510045**

**Batch : B2**

**1. Euclidean Algorithm**

The **Euclidean Algorithm** is a method for finding the greatest common divisor (GCD) of two integers. The GCD is the largest positive integer that divides both numbers without leaving a remainder. The algorithm is based on the principle that the GCD of two numbers also divides their difference.

**How It Works:**

1. Given two integers a and b (where a≥b ), divide a by b and find the remainder r.
2. Replace a with b and b with r.
3. Repeat the process until b becomes zero. The last non-zero remainder is the GCD.

**Example:**

* Find the GCD of 48 and 18.
* The GCD is 6, the last non-zero remainder.

#include <bits/stdc++.h>

using namespace std;

// Function to compute the GCD of two numbers using the Euclidean algorithm

int gcd(int a, int b)

{

    while (b != 0)

    {

        int remainder = a % b;

        a = b;

        b = remainder;

    }

    return a;

}

int main()

{

    int a, b;

    cout << "Enter two integers: ";

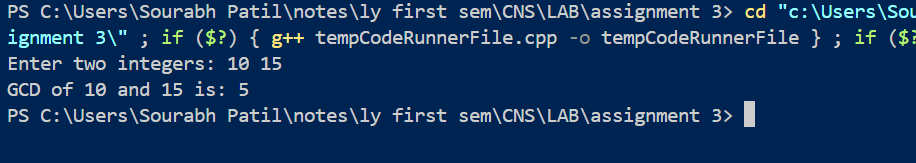
    cin >> a >> b;

    int result = gcd(a, b);

    cout << "GCD of " << a << " and " << b << " is: " << result << endl;

    return 0;

}

****

**2. Extended Euclidean Algorithm**

The **Extended Euclidean Algorithm** not only computes the GCD of two integers a and b, but it also finds integers x and y such that:

ax+by=GCD(a,b)

This is particularly useful in number theory, especially in solving linear Diophantine equations and finding modular inverses.

**How It Works:**

1. Perform the steps of the Euclidean algorithm while keeping track of coefficients xxx and yyy.
2. Start with initial values:
3. During each iteration of the Euclidean algorithm, update the coefficients using the relationship:

#include <iostream>

using namespace std;

// Function to implement the Extended Euclidean Algorithm

int extendedGCD(int a, int b, int &x, int &y)

{

    if (b == 0)

    {

        x = 1;

        y = 0;

        return a;

    }

    int x1, y1;

    int gcd = extendedGCD(b, a % b, x1, y1);

    x = y1;

    y = x1 - (a / b) \* y1;

    return gcd;

}

int main()

{

    int a, b;

    int x, y;

    cout << "Enter two integers: ";

    cin >> a >> b;

    int gcd = extendedGCD(a, b, x, y);

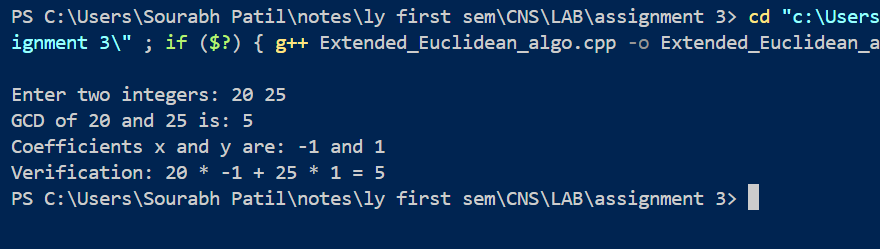
    cout << "GCD of " << a << " and " << b << " is: " << gcd << endl;

    cout << "Coefficients x and y are: " << x << " and " << y << endl;

    cout << "Verification: " << a << " \* " << x << " + " << b << " \* " << y << " = " << gcd << endl;

    return 0;

}

****