Name: Ritesh Pawar

Batch: B4

Subject: CNS Lab

PRN: 2020BTECS00068

**Aim: Find the GCD of two given numbers using Extended Euclidean Algorithm:**

**Theory:**

In arithmetic and computer programming, the extended Euclidean algorithm is an extension to the Euclidean algorithm, and computes, in addition to the greatest common divisor (gcd) of integers a and b, also the coefficients of Bézout's identity, which are integers x and y such that. The extended Euclidean algorithm also refers to a very similar algorithm for computing the polynomial greatest common divisor and the coefficients of Bézout's identity of two univariate polynomials.

**Code:**

#include <bits/stdc++.h>

using namespace std;

int gcdExtended(int a, int b, int \*x, int \*y) {

    // Initialize the table header

    cout << "Step   a   b   q   r   x   y" << endl;

    int x1, y1;  // To store results of recursive call

    int step = 1; // Initialize step counter

    int gcd;

    while (a != 0) {

        int q = b / a;

        int r = b % a;

        \*x = \*x - q \* x1;

        \*y = \*y - q \* y1;

        // Print the step details

        cout << step << "      " << a << "   " << b << "   " << q << "   " << r << "   " << \*x << "   " << \*y << endl;

        b = a;

        a = r;

        x1 = \*x;

        y1 = \*y;

        step++;

    }

    gcd = b; // GCD is stored in 'b'

    return gcd;

}

int main() {

    int x, y, a = 35, b = 15;

    int g = gcdExtended(a, b, &x, &y);

    cout << "GCD(" << a << ", " << b << ") = " << g << endl;

    return 0;

}

**Output:**



