**Cryptography and Network Security Lab**

**Name: Sumit Narake**

**PRN: 2020BTECS00023**

**Batch: B2**

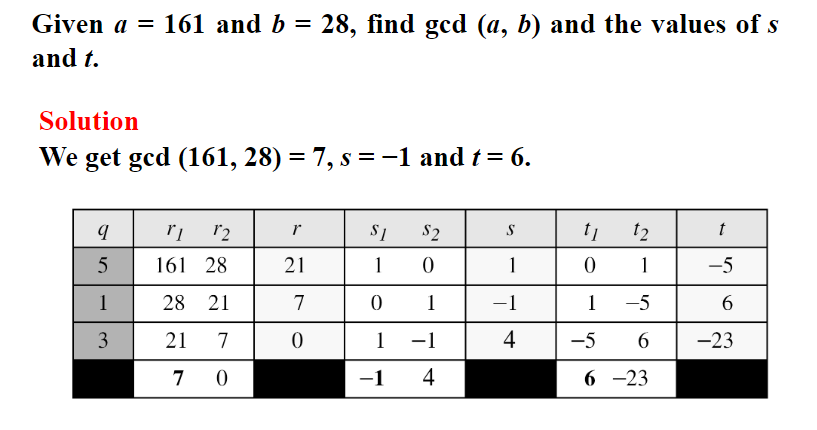
**EUCLIDIAN ALGORITHM**

**Aim:**

Implementation of Extended Euclidian algorithm.

**Theory:**

Examples:



**Code:**

#include <bits/stdc++.h>

using namespace std;

int ansS, ansT;

int findGcdExtended(int r1, int r2, int s1, int s2, int t1, int t2)

{

    // Base Case

    if (r2 == 0)

    {

        ansS = s1;

        ansT = t1;

        return r1;

    }

    int q = r1 / r2;

    int r = r1 % r2;

    int s = s1 - q \* s2;

    int t = t1 - q \* t2;

    cout << q << " " << r1 << " " << r2 << " " << r << " " << s1 << " " << s2 << " " << s << " " << t1 << " " << t2 << " " << t << endl;

    return findGcdExtended(r2, r, s2, s, t2, t);

}

int main()

{

    int num1, num2;

    cout << "\n Enter 1st number : ";

    cin >> num1;

    cout << "\n Enter 2nd number : ";

    cin >> num2;

    cout<<endl<< "q r1 r2 r s1 s2 s t1 t2 t" << endl;

    int gcd = findGcdExtended(num1, num2, 1, 0, 0, 1);

    cout <<endl<< "GCD is " << gcd << endl;

    cout <<endl<< "Value of s : "<<ansS << " " <<"Value of t : "<<ansT << endl;

    return 0;

}

**Output:**

