ISHITA HARDASMALANI
C14-2103058

EXPERIMENT NO. 1

Aim : Implementation of Extended Euclidean Algorithm

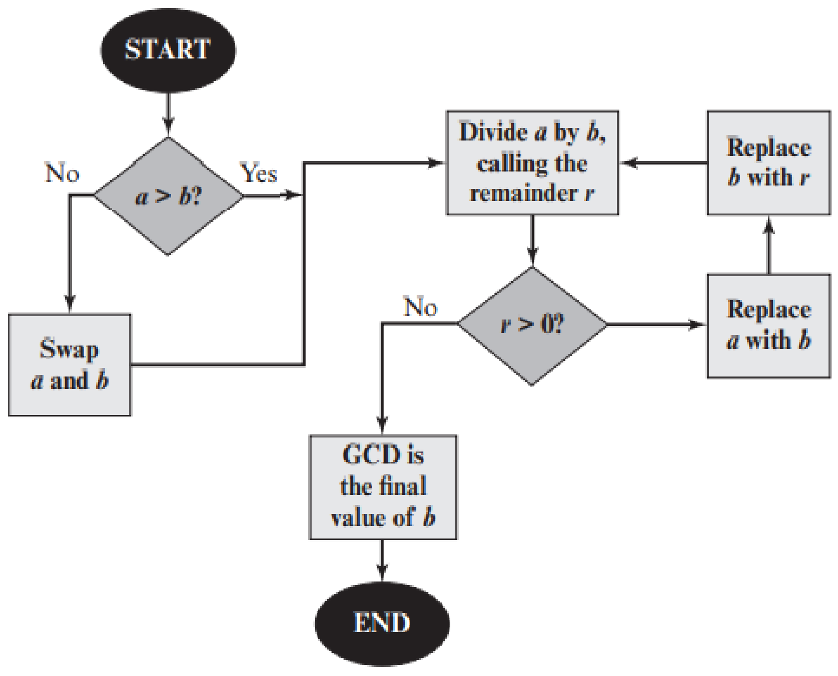
Theory:

The Euclidean Algorithm is an ancient and fundamental algorithm for finding the greatest common divisor (GCD) of two integers. It was named after the ancient Greek mathematician Euclid, who first described it in his Elements (circa 300 BCE). The GCD of two integers is the largest positive integer that divides both numbers without leaving a remainder.

Here's a simple version of the Euclidean Algorithm for finding the GCD of two integers, let's say a and b:

1. If b=0, then the GCD is a.
2. Otherwise, the GCD is the GCD of b and mod $a\backslash \text{mod} b$ (where \mod denotes the remainder after division).

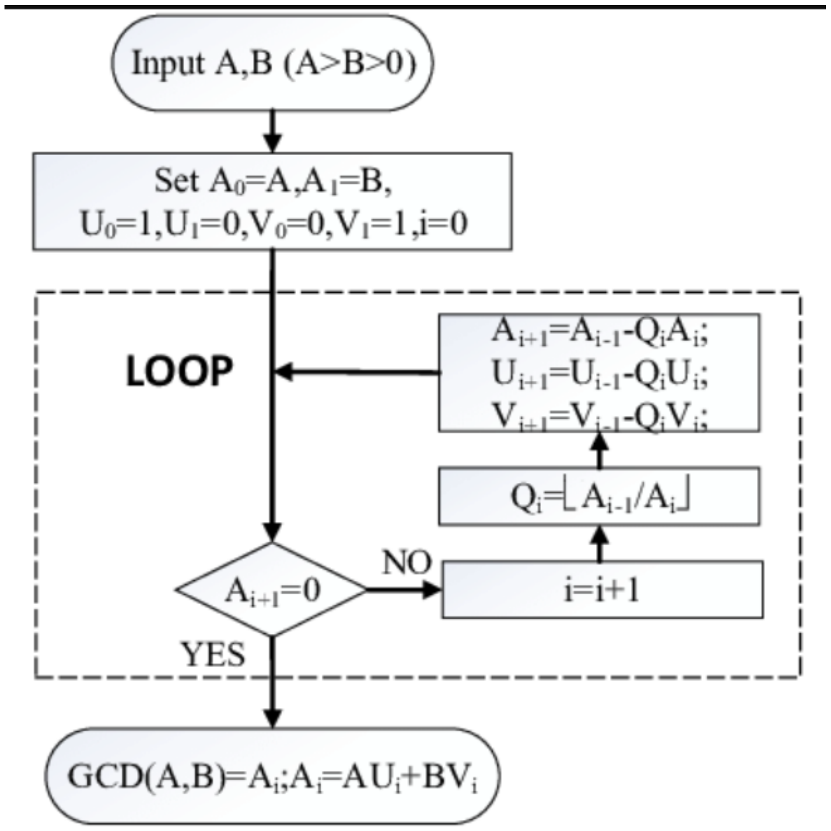
This algorithm continues to apply step 2 iteratively until b becomes 0, at which point the GCD is found.



The Extended Euclidean Algorithm is an extension of the Euclidean Algorithm that not only finds the GCD but also expresses it as a linear combination of the given numbers. It is particularly useful in solving linear Diophantine equations, which are equations of the form $ax+by=c$ where a, b, and c are integers.

The Extended Euclidean Algorithm works as follows:

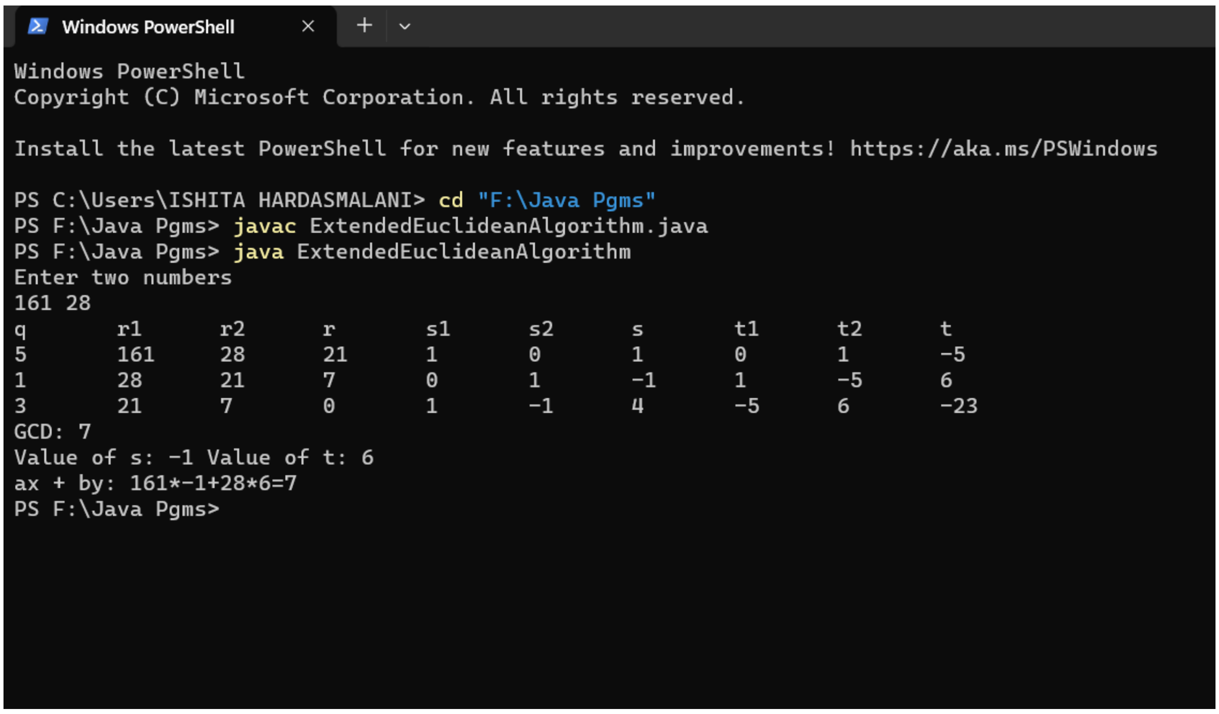
1. Apply the Euclidean Algorithm to find the GCD of a and b.
2. Express the GCD as a linear combination of a and b, i.e., find integers x and y such that $ax+by=\text{GCD}(a,b)$.


Input A,B (A>B>0)
Set $A_0=A, A_1=B$, $U_0=1, U_1=0, V_0=0, V_1=1, i=0$
LOOP
$A_{i+1}=A_{i-1}-Q_iA_i$; $U_{i+1}=U_{i-1}-Q_iU_i$; $V_{i+1}=V_{i-1}-Q_iV_i$;
$Q_i=\lfloor A_{i-1}/A_i \rfloor$
$A_{i+1}=0$
NO
$i=i+1$
YES
$\text{GCD}(A,B)=A_i; A_i=AU_i+BV_i$

Code:

```
import java.io.*;

import java.util.*;

public class ExtendedEuclideanAlgorithm {

        public static void main(String[] args) {

     Scanner sc = new Scanner(System.in);

     int a, b;

     System.out.println("Enter two numbers");

     a = sc.nextInt();

     b = sc.nextInt();

     if (a < b) {

        int temp = a;

        a = b;

        b = temp;

     }

     int r = a % b, q = a / b, r1 = a, r2 = b, s1 = 1, s2 = 0, t1 = 0, t2 = 1;

     int t = t1 - q * t2;

     int s = s1 - q * s2;

     System.out.println("q\tr1\tr2\tr\ts1\ts2\ts\tt1\tt2\tt\t");

     while (r != 0) {

    System.out.println(q + "\t" + r1 + "\t" + r2 + "\t" + r + "\t" + s1 + "\t" + s2 + "\t" + s +
"\t" + t1 + "\t"

            + t2 + "\t" + t);

        r1 = r2;

        r2 = r;

        r = r1 % r2;

        q = r1 / r2;

        s1 = s2;

        s2 = s;

        t1 = t2;

        t2 = t;
```

```
            t = t1 - q * t2;

            s = s1 - q * s2;

        }

        System.out.println(q + "\t" + r1 + "\t" + r2 + "\t" + r + "\t" + s1 + "\t" + s2 + "\t" + s +
"\t" + t1 + "\t"

            + t2 + "\t" + t);

        System.out.println("GCD: " + r2);

        System.out.println("Value of s: " + s2 + " Value of t: " + t2);

        int ans = a * s2 + b * t2;

        System.out.println("ax + by: " + a + "*" + s2 + "+" + b + "*" + t2 + "=" + ans);

    }

}
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ISHITA HARDASMALANI> cd "F:\Java Pgms"
PS F:\Java Pgms> javac ExtendedEuclideanAlgorithm.java
PS F:\Java Pgms> java ExtendedEuclideanAlgorithm
Enter two numbers
161 28
q       r1      r2      r       s1      s2      s       t1      t2      t
5       161     28      21      1       0       1       0       1       -5
1       28      21      7       0       1       -1      1       -5      6
3       21      7       0       1       -1      4       -5      6       -23
GCD: 7
Value of s: -1 Value of t: 6
ax + by: 161*-1+28*6=7
PS F:\Java Pgms>
```