C&NS Lab Assignment 9

Onkar Santosh Gavali (2019BTECS00037)

Batch B2

# Index

Euclidean and Extended Euclidean

* Explain Euclidean and Extended Euclidean
* Implement Euclidean and Extended Euclidean

# 

# 

# 

# 

# Euclidean and Extended Euclidean

Euclidean and Extended Euclidean used to calculate gcd, mutiplicative inverse and in many other mathematical sections and in cryptography.

Code:

#include <bits/stdc++.h>

using namespace std;

// Function for extended Euclidean Algorithm

long long gcdExtended(long long a, long long b, long long \*s1, long long s2,long long \*t1,long long t2) {

// Base Case

if (b == 0) {

cout << "# " << a << " " << b << " # | "

<< \*s1<<" "<<s2<<" # | "<<\*t1<<" "<<t2<<"#\n";

return a;

}

long long q=a/b;

long long s=\*s1-q\*s2,t=\*t1-q\*t2;

cout << q << " "

<< a << " "

<< b << " "

<< a % b<< " | "

<<\*s1<<" "

<<s2<<" "

<<s<<" | "

<<\*t1<<" "

<<t2<<" "

<<t<<endl;

\*s1=s2;

\*t1=t2;

long long gcd = gcdExtended(b, a % b, s1, s, t1, t);

return gcd;

}

int main() {

cout << "\t Euclidean and Extended Euclidean\n";

long long s = 1, t = 0,a,b;

cout << "\t Enter a and b to find GCD :\n";

//cout << "\t a = ";

cin >> a;

//cout << "\t b = ";

cin >> b;

cout << "q"

<< " "

<< "a"

<< " "

<< "b"

<< " "

<< "r"

<< " | "

<< "s1"

<< " "

<< "s2"

<< " "

<< "s"

<< " | "

<< "t1"

<< " "

<< "t2"

<< " "

<< "t\n";

long long res = gcdExtended(a, b, &s,0,&t,1);

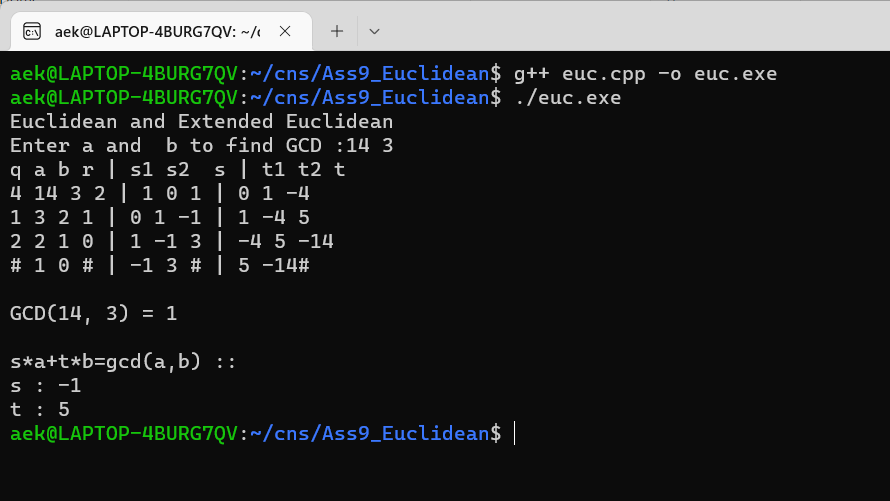
cout << "\nGCD(" << a << ", " << b << ") = " << res << endl;

cout<<"\ns\*a+t\*b=gcd(a,b) :: \n"<<"s : "<<s<<"\nt : "<<t<<endl;

return 0;

}

Output:



# 

# 