"Experiment 1.1"

Student Name: **SUMIT KUMAR** UID: **20BCS8226**
Branch: **CSE** Section/Group: **808-A**
Semester: **5** Date of Performance: **04-08-22**
Subject Name: **Design and Analysis of Algorithms Lab** Subject Code: **20CSP-312**

1. Aim/Overview of the practical:

Code and analyse to compute the greatest common divisor (GCD) of two numbers.

2. Algorithm/Flowchart (For programming based labs):

START

Step 1: Let a, b be the two numbers

Step 2: a mod b = c

Step 3: Let a = b and b = c

Step 4: Repeat Steps 2 and 3 until a mod b is greater than 0

Step 5: GCD = b

END

4. Steps for experiment/practical/Code:

```
#include <bits/stdc++.h>
using namespace std;

int gcd(int a, int b){
    if(a%b==0) return b;
    else{
        int c = a%b;
        a=b;
        b=c;
    }
    return b;
}

int main()
{
    int a, b; cin>>a>>b;
    cout<<"GCD of "<<a<<" and "<<b<<" is : "<<gcd(a,b);

    return 0;
}
```

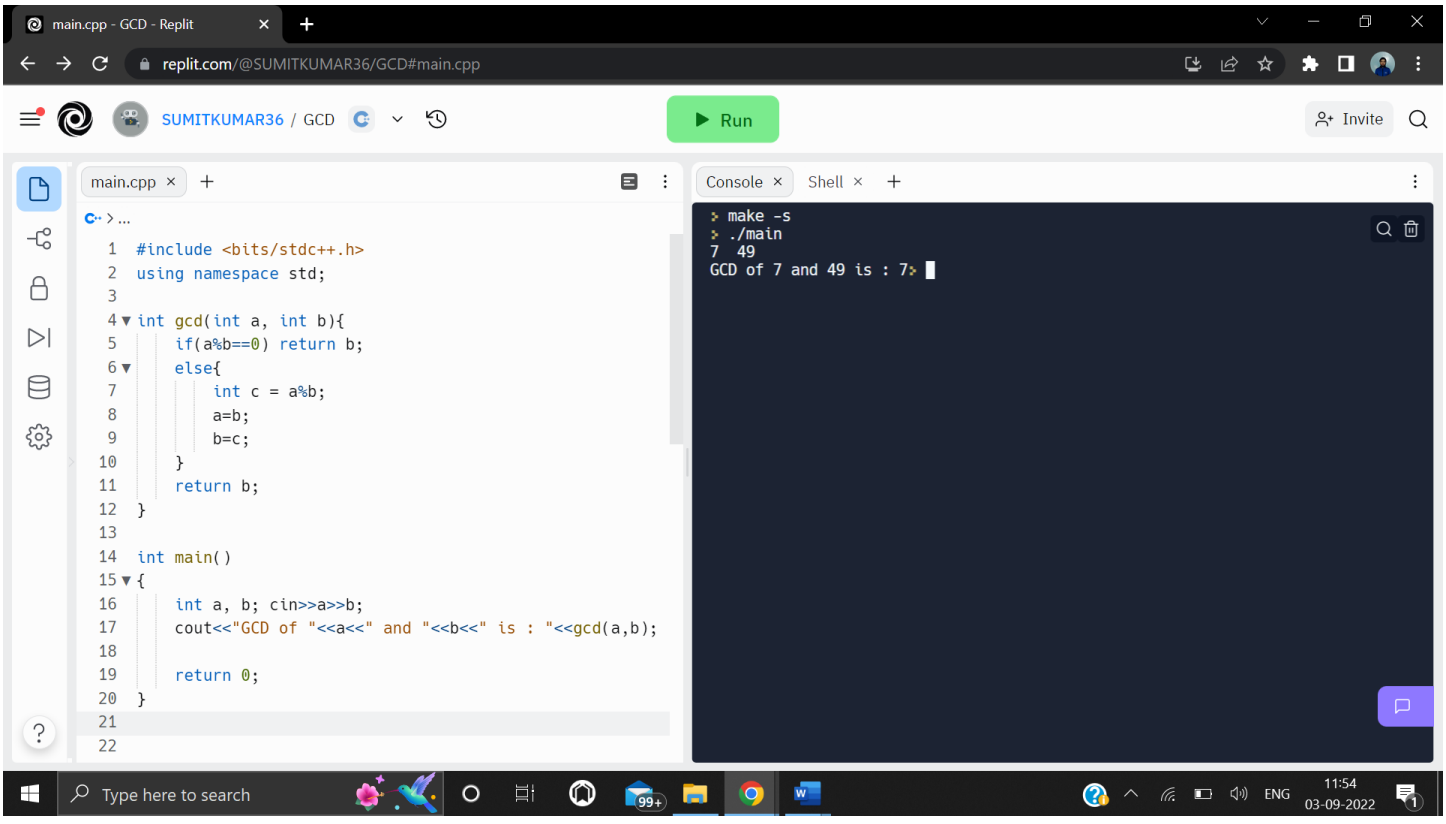
5. Observations/Discussions/ Complexity Analysis:

If we subtract a smaller number from a larger one (we reduce a larger number), GCD doesn't change. So, if we keep subtracting repeatedly the larger of two, we end up with GCD.

Now instead of subtraction, if we divide the smaller number, the algorithm stops when we find the remainder 0.

The time and space complexity of this algorithm is O(log(smaller number)).

6. Result/Output/Writing Summary:



Learning outcomes (What I have learnt):

1. Learnt efficient algorithm of calculating GCD using Euclid's method.
2. Learnt how to code Euclid's algorithm in C++.
3. Learnt how to analyze the complexity of recursive algorithms like Euclid's.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			