

Experiment 1

Student Name: Rohan Chauhan

UID: 20BCS3311

Branch: B.E. CSE

Section/Group: 20BCS-WM_701/B

Semester: 5th

Date of Performance: 08/08/2022

Subject Name: Design and Analysis of Algorithms

Subject Code: 20CSP-312

1. Aim/Overview of the practical:

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

2. Task to be done/ Which logistics used:

The GCD of two or more integers is the largest integer that divides each of the integers such that their remainder is 0 or we say that the GCD or HCF of two numbers is the largest number that divides both of them.

GCD of 20,30 = 10 (10 is the largest number which divides 20 and 30 with the remainder 0)

GCD of 42,120,285 = 3 (3 is the largest number which divides 42,120 and 285 with remainder 0)

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

Euclidean Algorithm for Greatest common divisor (GCD)-

An efficient solution is to use Euclidean Algorithm which is the main algorithm used for this purpose.

The idea is, that the GCD of two numbers doesn't change if the smaller number is subtracted from the bigger number.

The Euclidean algorithm finds the GCD of two numbers. You will better understand the algo by seeing it in action

Assuming you want to calculate the GCD of 1220 and 516, let's apply the Euclidean Algorithm.

4. Steps for experiment/practical/Code:

```
#include <iostream>
using namespace std;
int GCD(int a, int b)
{
    if (a == 0)
    {
        return b;
    }
    if (b == 0)
    {
        return a;
    }
    int temp;
    if (a > b)
    {
        temp = a;
        a = b;
        b = temp;
    }
    b = b % a;
    if (b != 0)
    {
        a = GCD(b, a);
    }
    return a;
}

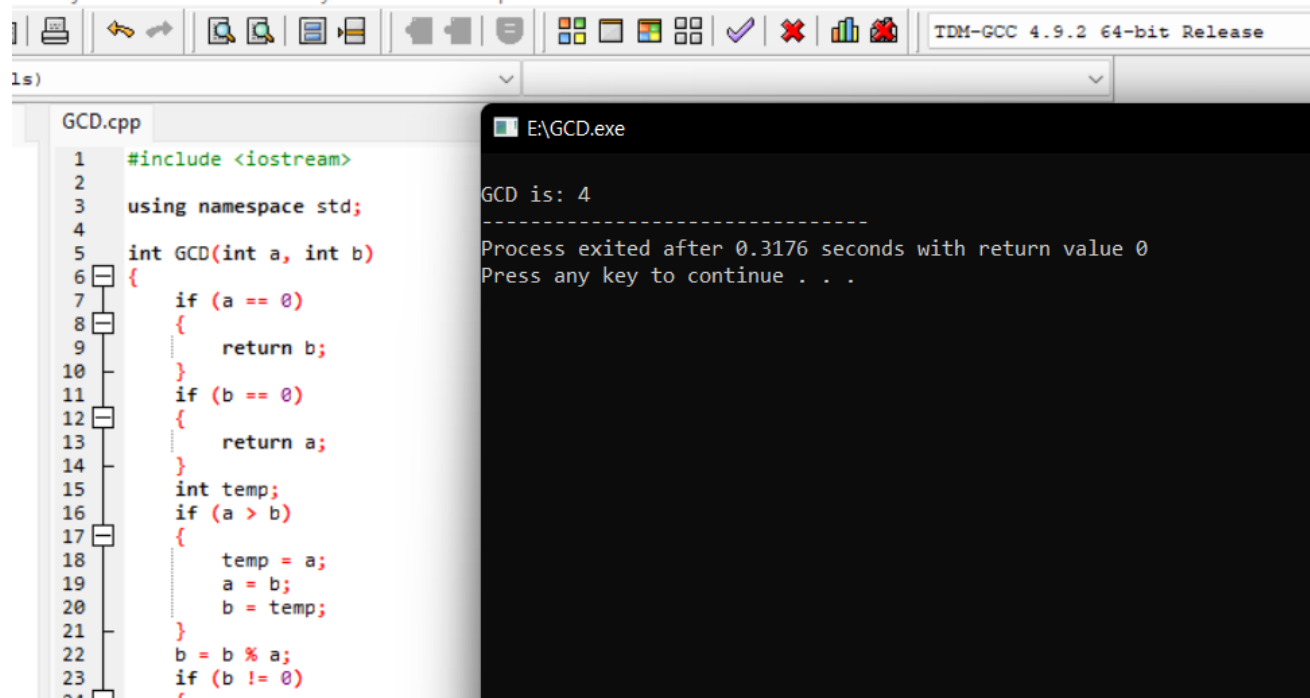
int main()
{
    int gcd = GCD(1220, 516);
```

```
cout << "\nGCD is: "<<gcd;  
}
```

5. Observations/Discussions/ Complexity Analysis:

g] - Dev-C++ 5.11

v Project Execute Tools AStyle Window Help



The screenshot displays the Dev-C++ 5.11 IDE. The left pane shows the source code for `GCD.cpp`, which implements a function to calculate the Greatest Common Divisor (GCD) of two integers `a` and `b`. The code includes `<iostream>`, uses the `std` namespace, and defines the `GCD` function. It handles edge cases where `a` or `b` is zero and uses a loop to find the GCD by repeatedly dividing the larger number by the smaller one. The right pane shows the execution output for `E:\GCD.exe`, which prints "GCD is: 4", followed by a separator line, the execution time "Process exited after 0.3176 seconds with return value 0", and a prompt "Press any key to continue . . .".

```
ls)
GCD.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int GCD(int a, int b)
6  {
7      if (a == 0)
8      {
9          return b;
10     }
11     if (b == 0)
12     {
13         return a;
14     }
15     int temp;
16     if (a > b)
17     {
18         temp = a;
19         a = b;
20         b = temp;
21     }
22     b = b % a;
23     if (b != 0)
24     {
25         return GCD(a, b);
26     }
27     return a;
28 }
29
30 int main()
31 {
32     int a, b;
33     cout << "Enter two numbers: ";
34     cin >> a >> b;
35     cout << "\nGCD is: "<<GCD(a, b);
36     return 0;
37 }
```

```
E:\GCD.exe
GCD is: 4
-----
Process exited after 0.3176 seconds with return value 0
Press any key to continue . . .
```