

1. Briefly explain Greatest Common Divisor (GCD).

The greatest common divisor also known as the highest common factor (HCF) is a mathematical concept used to find the largest positive integer that divides two or more numbers without leaving a remainder.

2. Explain the steps of the Euclidean Algorithm.

1. Start with two given integers, let's call them "a" and "b", where "a" is typically the larger number.
2. Divide "a" by "b" and take the remainder. Let's call this remainder "r".
3. If the remainder "r" is zero, then the algorithm stops and the GCD is equal to the value of "b". In this case, "b" is the largest number that divides both "a" and "b" without leaving a remainder.
4. If the remainder "r" is not zero, then set "a" equal to "b" and "b" equal to "r".
5. Repeat steps 2 to 4 using the updated values of "a" and "b".
6. Continue this process of dividing "a" by "b" and taking the remainder until the remainder becomes zero.
7. Once the remainder becomes zero, the algorithm stops, and the GCD is equal to the last non-zero remainder, which is the value of "b".

As parameters of the algorithm, it takes two numbers as number 1 and number 2 where number 1 is always the greater value.

If number 2 is equal to 0 the GCD value will be returned as number 1.

Else the GCD (num1, number2)  $\Rightarrow$  GCD (number2, GCD (number1 % number2)).

}(Greatest common factor) logic for 3307 coding

{(0 = 1 means) slider

: General if 1 means 2 equals first

BATMA

8. Write a Function Using Pseudo or Source code to find out the GCD using recursive.

```
public class RecursiveGCD {  
    public static void main(String[] args){  
        int num1 = 120;  
        int num2 = 35;  
        System.out.print("GCD(" + num1 + "," + num2 + ") = ");  
        System.out.print(gcd(num1, num2));  
    }  
    public static int gcd(int num1, int num2){  
        if(num2 == 0){  
            return num1;  
        }  
        return gcd(num2, num1 % num2);  
    }  
}
```

9. Try to use the iteration to get the same results.

```
public class IterativeGCD {  
    public static void main(String[] args){  
        int num1 = 120;  
        int num2 = 35;  
        System.out.print("GCD(" + num1 + "," + num2 + ") = ");  
        System.out.print(gcd(num1, num2));  
    }  
    public static int gcd(int num1, int num2){  
        while(num2 != 0){  
            int temp = num1 % num2;
```

Date \_\_\_\_\_

CA

SAD

WEB

e GCD

```
num1 = num2;  
num2 = temp;  
}  
return num1;  
}
```

5. What is defined by prime factorization.

Finding out the the prime numbers that divided & given number without leaving a remainder.

6. Geographically represent how to identify the prime factorization.

Ex:

