75 Days of Code

Day2
Problem no: 1071

For two strings s and t, we say "t divides s" if and only if s = t + ... + t (i.e., t is concatenated with itself one or more times).

Given two strings str1 and str2, return *the largest string* x *such that* x *divides both* str1 *and* str2.

Example 1:

Input: str1 = "ABCABC", str2 = "ABC"
Output: "ABC"

Example 2:

Input: str1 = "ABABAB", str2 = "ABAB"
Output: "AB"

Example 3:

Input: str1 = "LEET", str2 = "CODE"
Output: ""

Solution

For this problem ,

Assuming str1 and str2 have a common divisor , str1 = t1+t1... *m times and str2 =t1+t1... *ntimes , if we combine them the resultant string should be equal so str1+str2 = str2+str1 , so to have gcd of string this condition should be true .

Shubham Agrahari

To calculate gcd i am using **Euclidean algorithm**

It is an efficient method for calculating gcd of two numbers

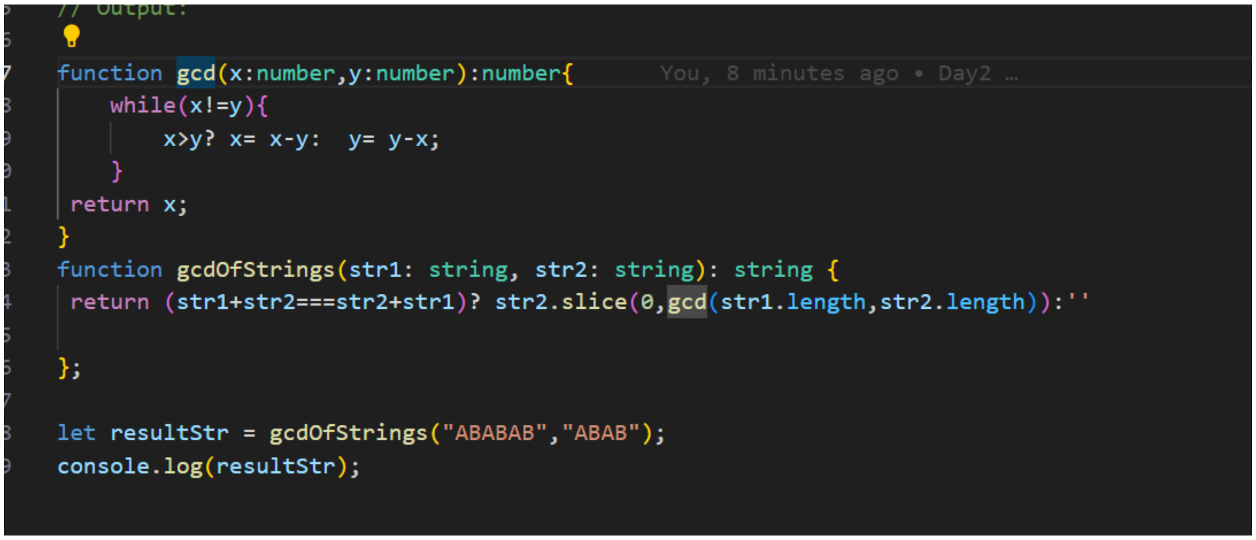
For eg two find gcd of two numbers 12 18

Start iteration with 12 and 18 . here we subtract smaller number from larger until we get a number equal to both 12 and 18

code

```
   function gcd(x,y){
   while(x!=y){
       if(x>y){
      x = x-y;
  }else {
     y= y-x;
 }
  return x;


}
```



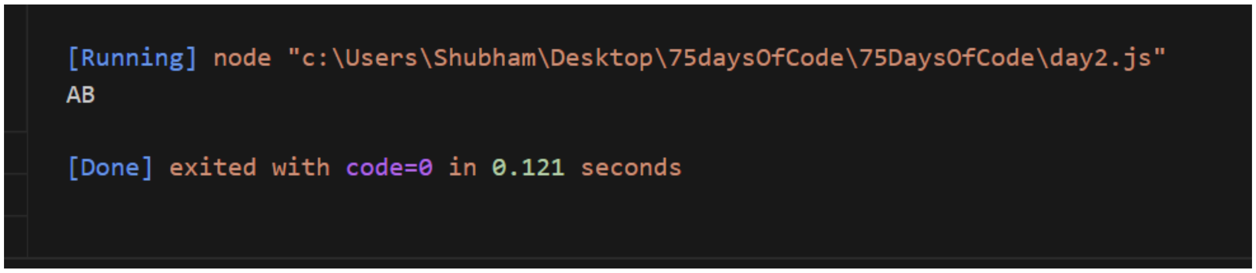
```
// Output:

function gcd(x:number,y:number):number{        You, 8 minutes ago • Day2 …
    while(x!=y){
        x>y? x= x-y:  y= y-x;
    }
 return x;
}
function gcdOfStrings(str1: string, str2: string): string {
 return (str1+str2===str2+str1)? str2.slice(0,gcd(str1.length,str2.length)):''

};

let resultStr = gcdOfStrings("ABABAB","ABAB");
console.log(resultStr);
```

Shubham Agrahari



```
[Running] node "c:\Users\Shubham\Desktop\75daysOfCode\75DaysOfCode\day2.js"
AB

[Done] exited with code=0 in 0.121 seconds
```

Shubham Agrahari