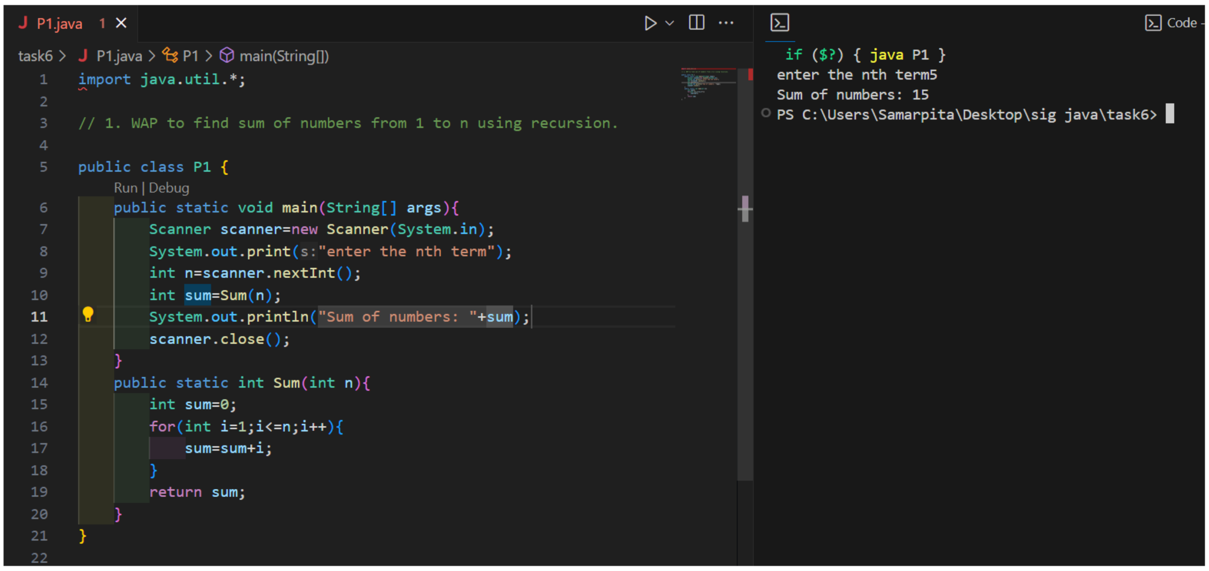
1. WAP to find sum of numbers from 1 to n using recursion.



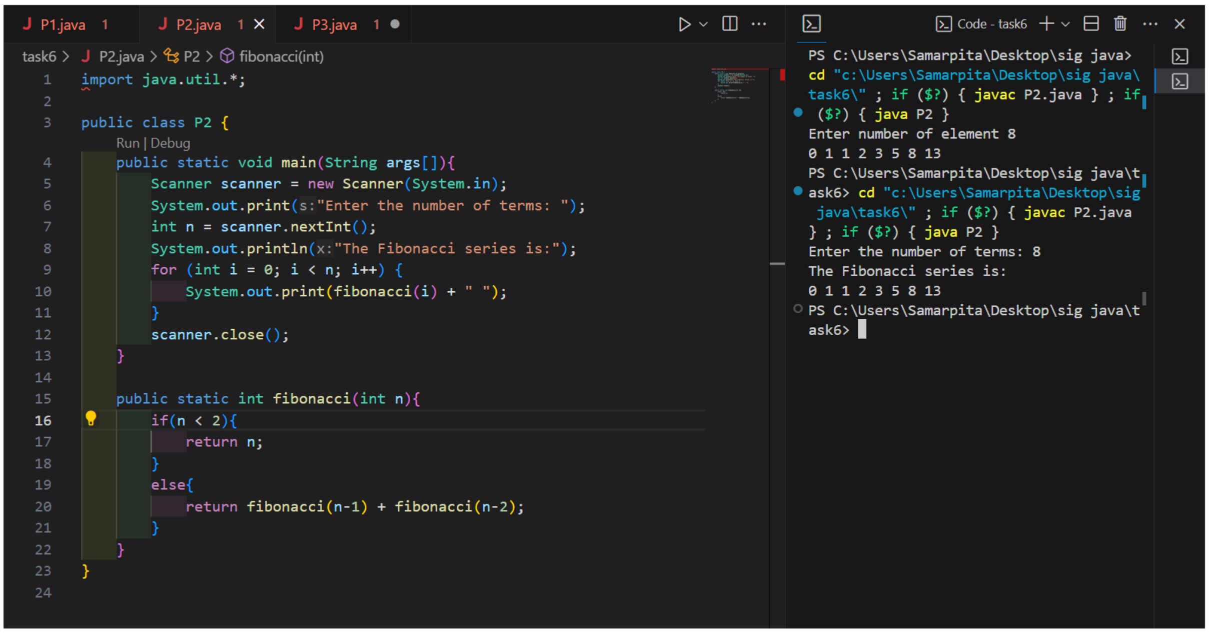
```java
import java.util.*;

// 1. WAP to find sum of numbers from 1 to n using recursion.

public class P1 {
    public static void main(String[] args){
        Scanner scanner=new Scanner(System.in);
        System.out.print(s:"enter the nth term");
        int n=scanner.nextInt();
        int sum=Sum(n);
        System.out.println("Sum of numbers: "+sum);
        scanner.close();
    }
    public static int Sum(int n){
        int sum=0;
        for(int i=1;i<=n;i++){
            sum=sum+i;
        }
        return sum;
    }
}
```

```
 if ($?) { java P1 }
enter the nth term5
Sum of numbers: 15
PS C:\Users\Samarpita\Desktop\sig java\task6> 
```

2. WAP to print the Fibonacci Series using recursion.



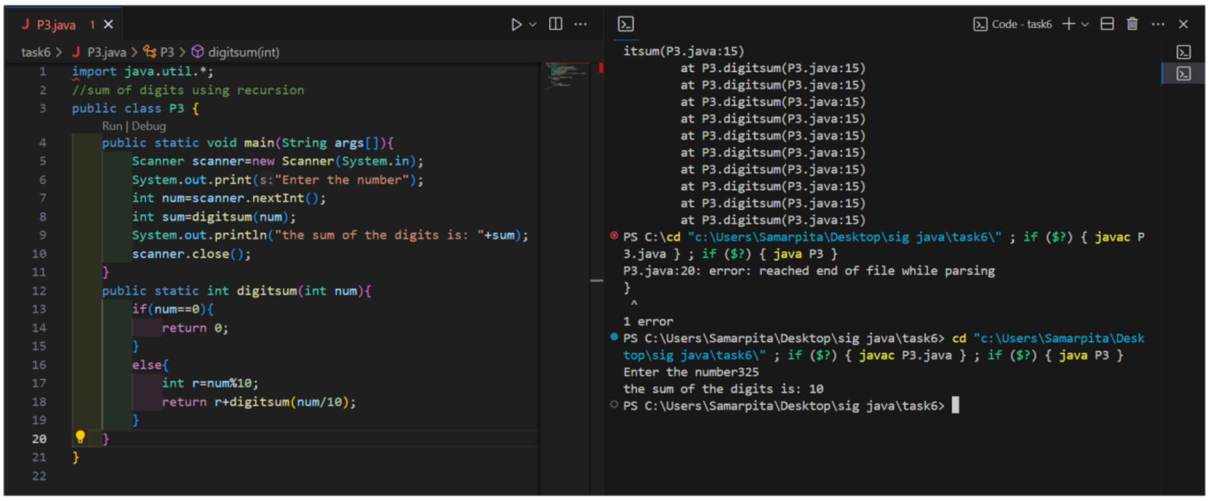
```java
import java.util.*;

public class P2 {
    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Enter the number of terms: ");
        int n = scanner.nextInt();
        System.out.println(x:"The Fibonacci series is:");
        for (int i = 0; i < n; i++) {
            System.out.print(fibonacci(i) + " ");
        }
        scanner.close();
    }

    public static int fibonacci(int n){
        if(n < 2){
            return n;
        }
        else{
            return fibonacci(n-1) + fibonacci(n-2);
        }
    }
}
```

```
PS C:\Users\Samarpita\Desktop\sig java>
cd "c:\Users\Samarpita\Desktop\sig java\
task6\" ; if ($?) { javac P2.java } ; if
 ($?) { java P2 }
Enter number of element 8
0 1 1 2 3 5 8 13
PS C:\Users\Samarpita\Desktop\sig java\t
ask6> cd "c:\Users\Samarpita\Desktop\sig
 java\task6\" ; if ($?) { javac P2.java
} ; if ($?) { java P2 }
Enter the number of terms: 8
The Fibonacci series is:
0 1 1 2 3 5 8 13
PS C:\Users\Samarpita\Desktop\sig java\t
ask6> 
```

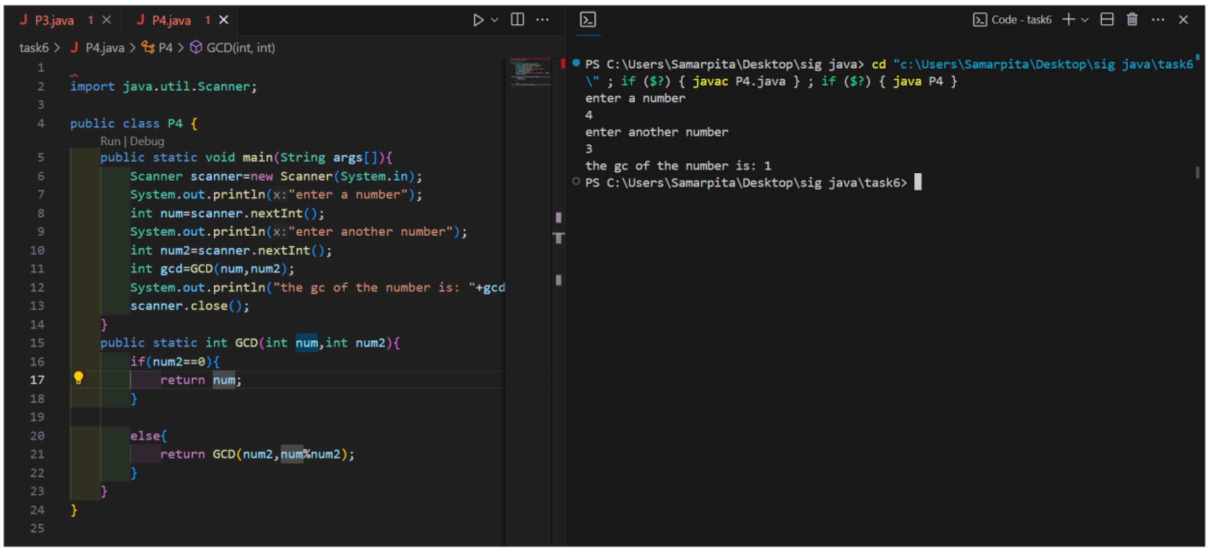
3. WAP to find the sum of digits of a number using recursion.



```java
import java.util.*;
//sum of digits using recursion
public class P3 {
    public static void main(String args[]){
        Scanner scanner=new Scanner(System.in);
        System.out.print(s:"Enter the number");
        int num=scanner.nextInt();
        int sum=digitsum(num);
        System.out.println("the sum of the digits is: "+sum);
        scanner.close();
    }
    public static int digitsum(int num){
        if(num==0){
            return 0;
        }
        else{
            int r=num%10;
            return r+digitsum(num/10);
        }
    }
}
```

```
itsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
        at P3.digitsum(P3.java:15)
PS C:\cd "c:\Users\Samarpita\Desktop\sig java\task6\" ; if ($?) { javac P
3.java } ; if ($?) { java P3 }
P3.java:20: error: reached end of file while parsing
}
 ^
1 error
PS C:\Users\Samarpita\Desktop\sig java\task6> cd "c:\Users\Samarpita\Desk
top\sig java\task6\" ; if ($?) { javac P3.java } ; if ($?) { java P3 }
Enter the number325
the sum of the digits is: 10
PS C:\Users\Samarpita\Desktop\sig java\task6>
```

4. WAP to find the gcd of two number using recursion.



```java

import java.util.Scanner;

public class P4 {
    public static void main(String args[]){
        Scanner scanner=new Scanner(System.in);
        System.out.println(x:"enter a number");
        int num=scanner.nextInt();
        System.out.println(x:"enter another number");
        int num2=scanner.nextInt();
        int gcd=GCD(num,num2);
        System.out.println("the gc of the number is: "+gcd
        scanner.close();
    }
    public static int GCD(int num,int num2){
        if(num2==0){
            return num;
        }

        else{
            return GCD(num2,num%num2);
        }
    }
}
```

```
PS C:\Users\Samarpita\Desktop\sig java> cd "c:\Users\Samarpita\Desktop\sig java\task6
\" ; if ($?) { javac P4.java } ; if ($?) { java P4 }
enter a number
4
enter another number
3
the gc of the number is: 1
PS C:\Users\Samarpita\Desktop\sig java\task6>
```

5. Write Advantages and Disadvantages of Recursion.

advantages	Disadvantages
Recursive solutions often provide simpler, more elegant solutions to certain problems, especially those that can be naturally expressed in terms of smaller instances of the same problem.	Recursive function calls involve additional overhead such as function call stack management and memory allocation, which can impact performance, especially for deeply nested recursion or large input sizes.
Recursive code can be easier to read and understand, especially for problems that involve repetitive tasks or patterns.	Recursive functions consume stack space for each function call, and if the recursion depth is too large, it can lead to a stack overflow error, causing the program to terminate unexpectedly.
Recursive solutions can sometimes lead to shorter and more concise code compared to iterative solutions, which can be beneficial for maintenance and readability.	Recursive code can be more challenging to debug compared to iterative code, especially when dealing with complex recursive calls or base cases.
Recursion promotes modularity by breaking down a problem into smaller, self-contained units (subproblems), which can make the code easier to manage and debug.	If not properly designed, recursive functions may enter an infinite loop, consuming system resources indefinitely and leading to program crashes.