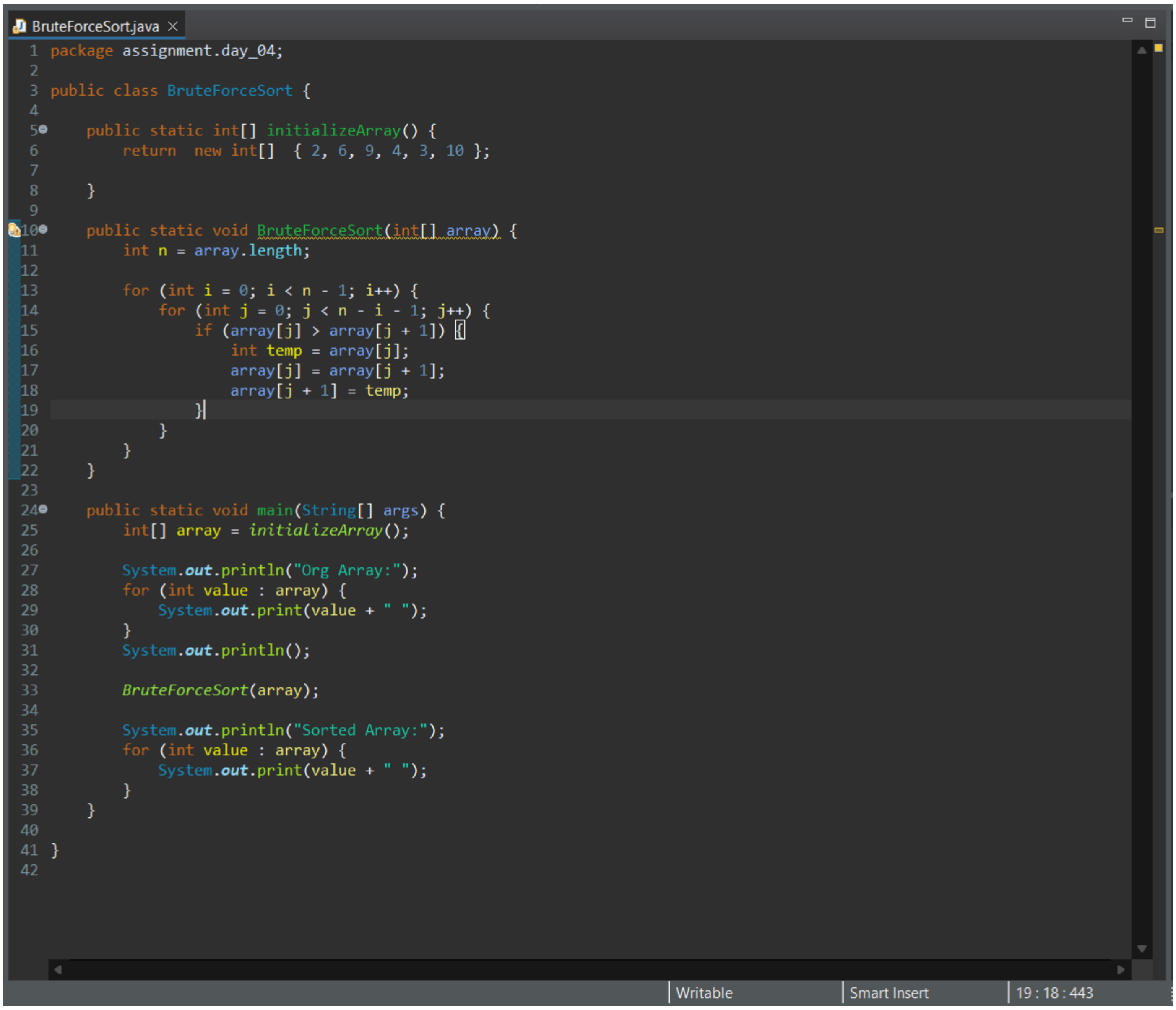
Java- Assignment (Day 4)

Task 1: Array Sorting and Searching

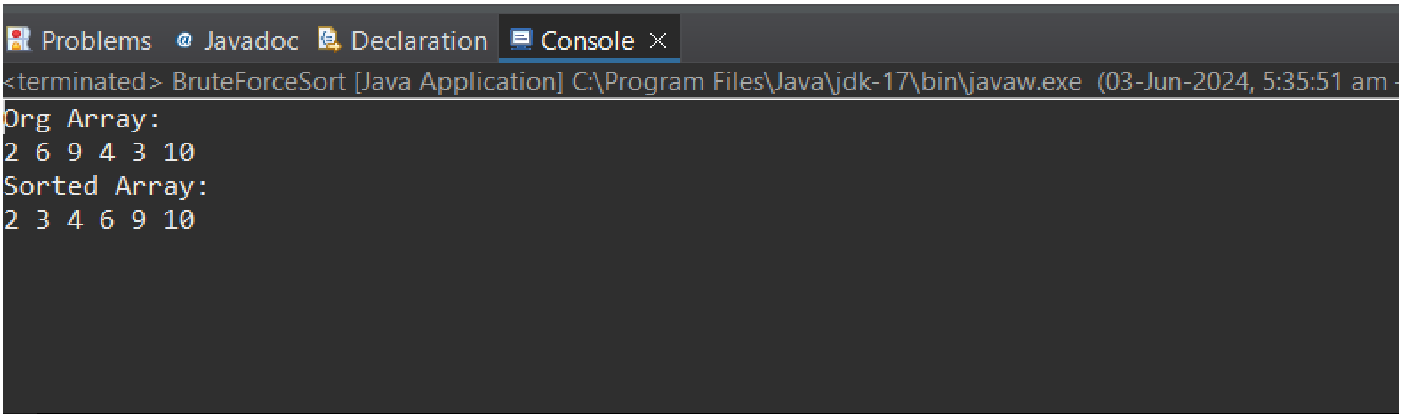
a) Implement a function called BruteForceSort that sorts an array using the brute force approach. Use this function to sort an array created with InitializeArray.

Code:



```java
package assignment.day_04;

public class BruteForceSort {

    public static int[] initializeArray() {
        return  new int[]  { 2, 6, 9, 4, 3, 10 };

    }

    public static void BruteForceSort(int[] array) {
        int n = array.length;

        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
        int[] array = initializeArray();

        System.out.println("Org Array:");
        for (int value : array) {
            System.out.print(value + " ");
        }
        System.out.println();

        BruteForceSort(array);

        System.out.println("Sorted Array:");
        for (int value : array) {
            System.out.print(value + " ");
        }
    }

}
```

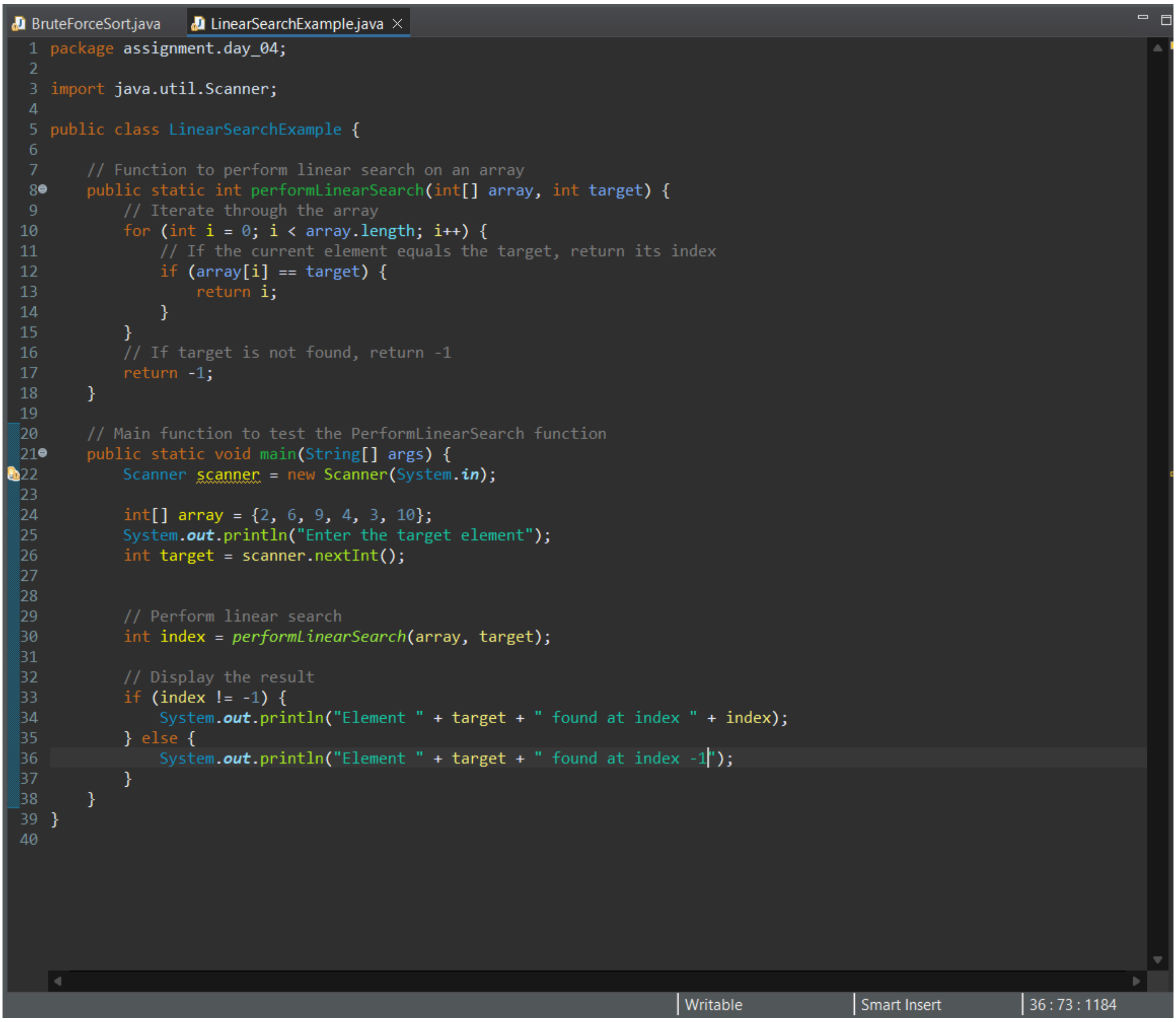
Output:



```
<terminated> BruteForceSort [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (03-Jun-2024, 5:35:51 am
Org Array:
2 6 9 4 3 10
Sorted Array:
2 3 4 6 9 10
```

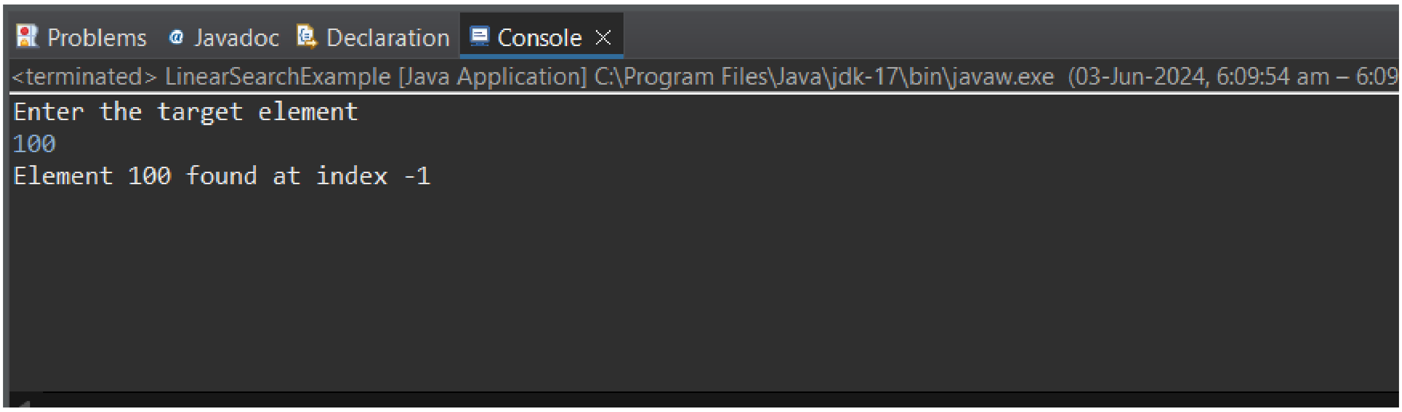
b) Write a function named PerformLinearSearch that searches for a specific element in an array and returns the index of the element if found or -1 if not found.

Code:



```java
package assignment.day_04;

import java.util.Scanner;

public class LinearSearchExample {

    // Function to perform linear search on an array
    public static int performLinearSearch(int[] array, int target) {
        // Iterate through the array
        for (int i = 0; i < array.length; i++) {
            // If the current element equals the target, return its index
            if (array[i] == target) {
                return i;
            }
        }
        // If target is not found, return -1
        return -1;
    }

    // Main function to test the PerformLinearSearch function
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int[] array = {2, 6, 9, 4, 3, 10};
        System.out.println("Enter the target element");
        int target = scanner.nextInt();


        // Perform linear search
        int index = performLinearSearch(array, target);

        // Display the result
        if (index != -1) {
            System.out.println("Element " + target + " found at index " + index);
        } else {
            System.out.println("Element " + target + " found at index -1");
        }
    }
}

```

Output:

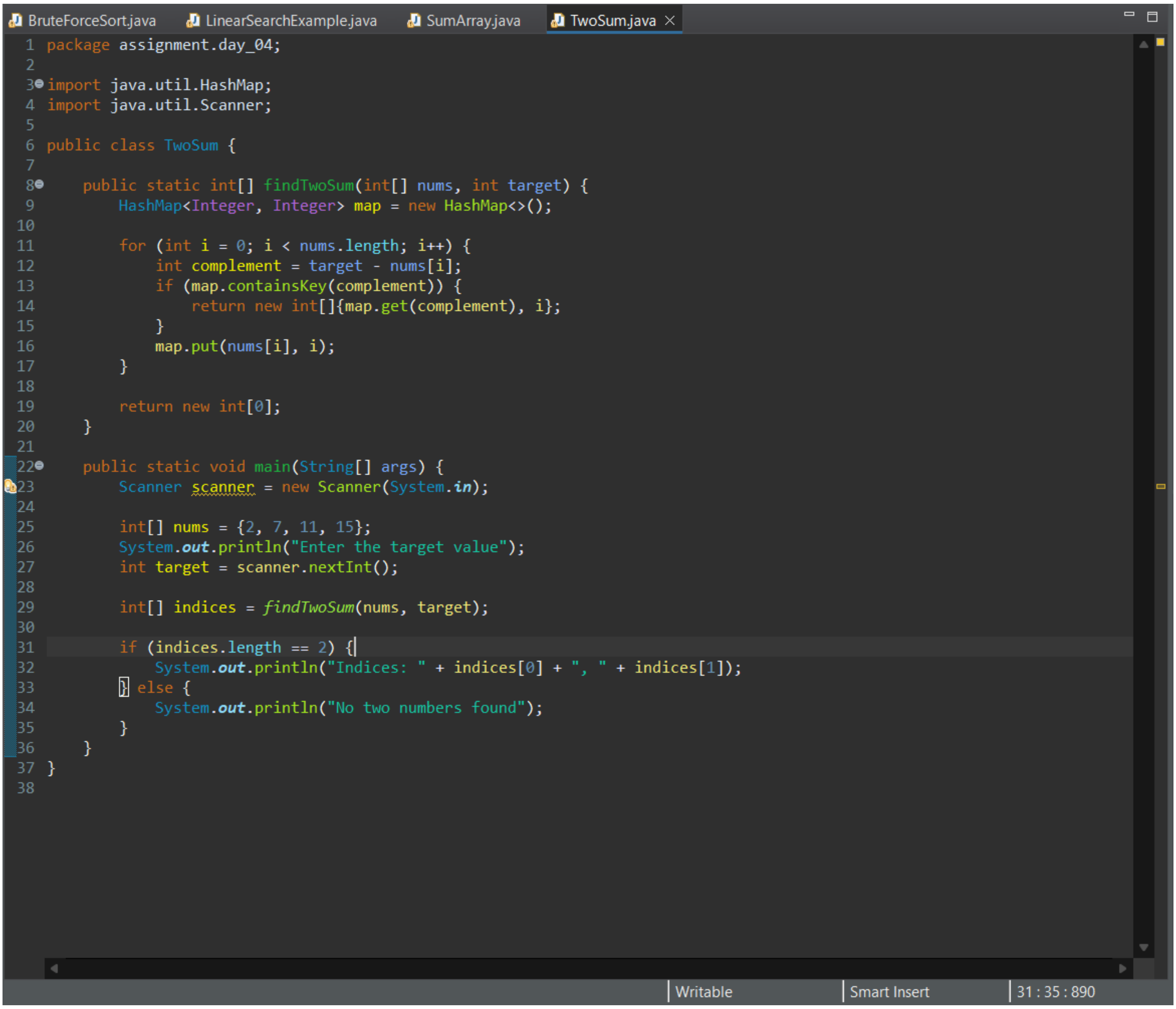


```
<terminated> LinearSearchExample [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (03-Jun-2024, 6:09:54 am – 6:09
Enter the target element
100
Element 100 found at index -1
```

Task 2: Two-Sum Problem

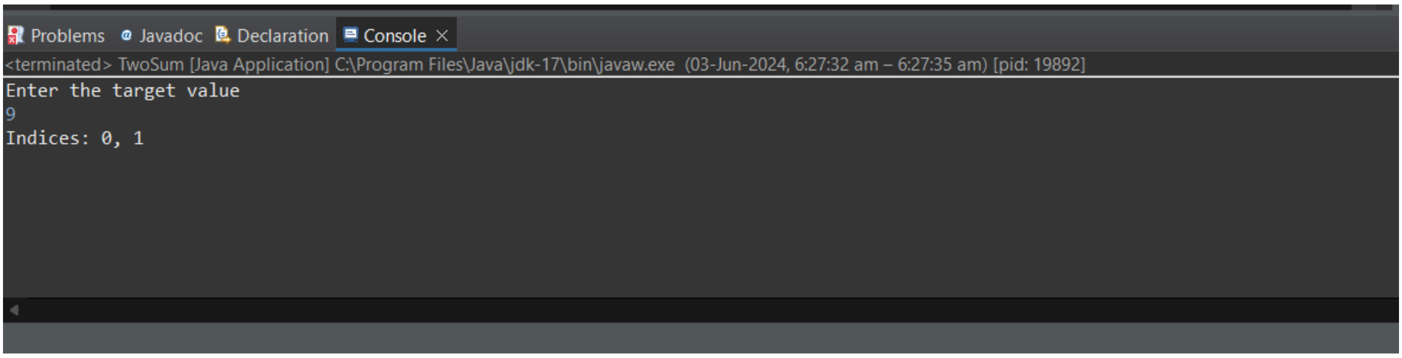
a) Given an array of integers, write a program that finds if there are two numbers that add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice. Optimize the solution for time complexity.

Code:



```java
package assignment.day_04;

import java.util.HashMap;
import java.util.Scanner;

public class TwoSum {

    public static int[] findTwoSum(int[] nums, int target) {
        HashMap<Integer, Integer> map = new HashMap<>();

        for (int i = 0; i < nums.length; i++) {
            int complement = target - nums[i];
            if (map.containsKey(complement)) {
                return new int[]{map.get(complement), i};
            }
            map.put(nums[i], i);
        }

        return new int[0];
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int[] nums = {2, 7, 11, 15};
        System.out.println("Enter the target value");
        int target = scanner.nextInt();

        int[] indices = findTwoSum(nums, target);

        if (indices.length == 2) {
            System.out.println("Indices: " + indices[0] + ", " + indices[1]);
        } else {
            System.out.println("No two numbers found");
        }
    }
}
```

Ouput:

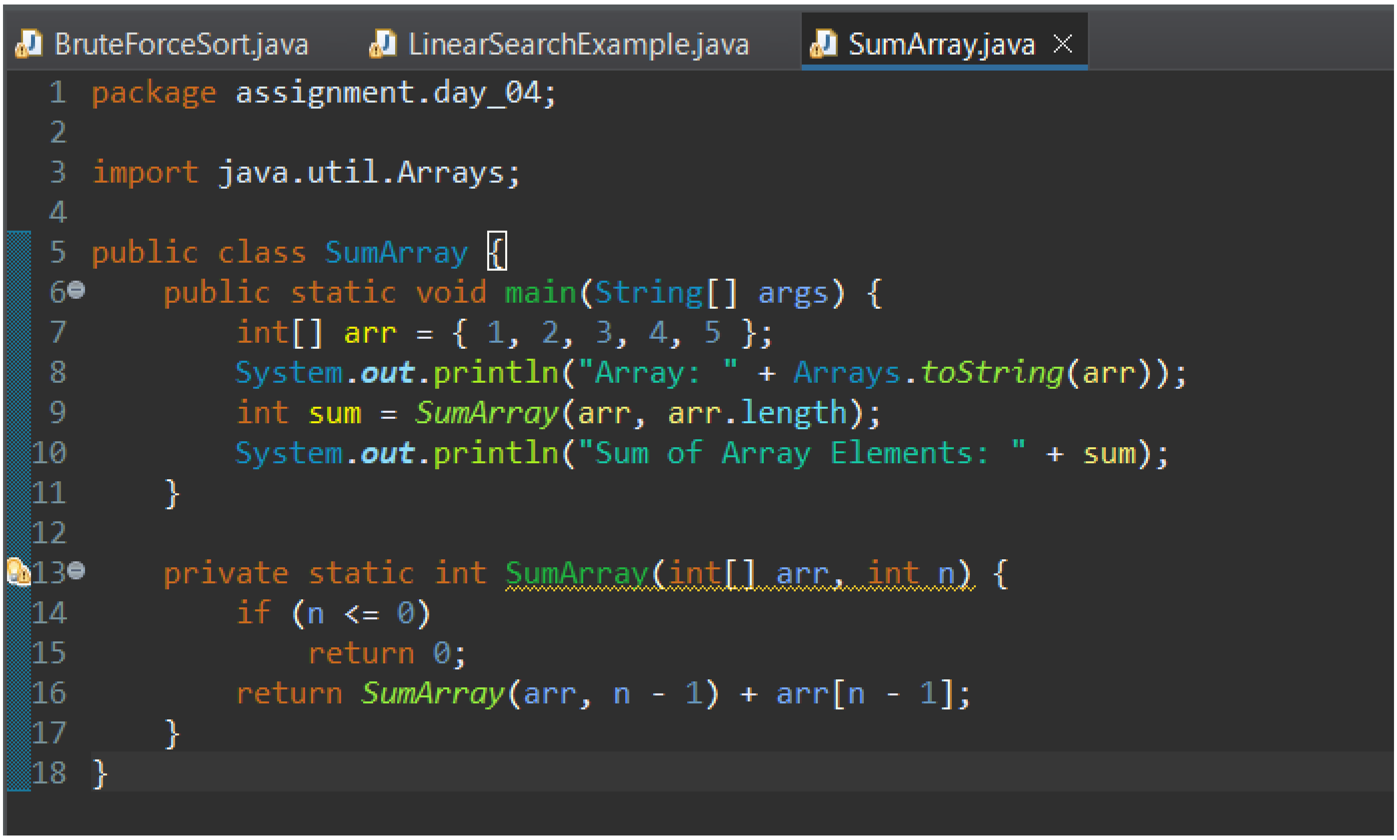


```
<terminated> TwoSum [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (03-Jun-2024, 6:27:32 am – 6:27:35 am) [pid: 19892]
Enter the target value
9
Indices: 0, 1
```

Task 3: Understanding Functions through Arrays

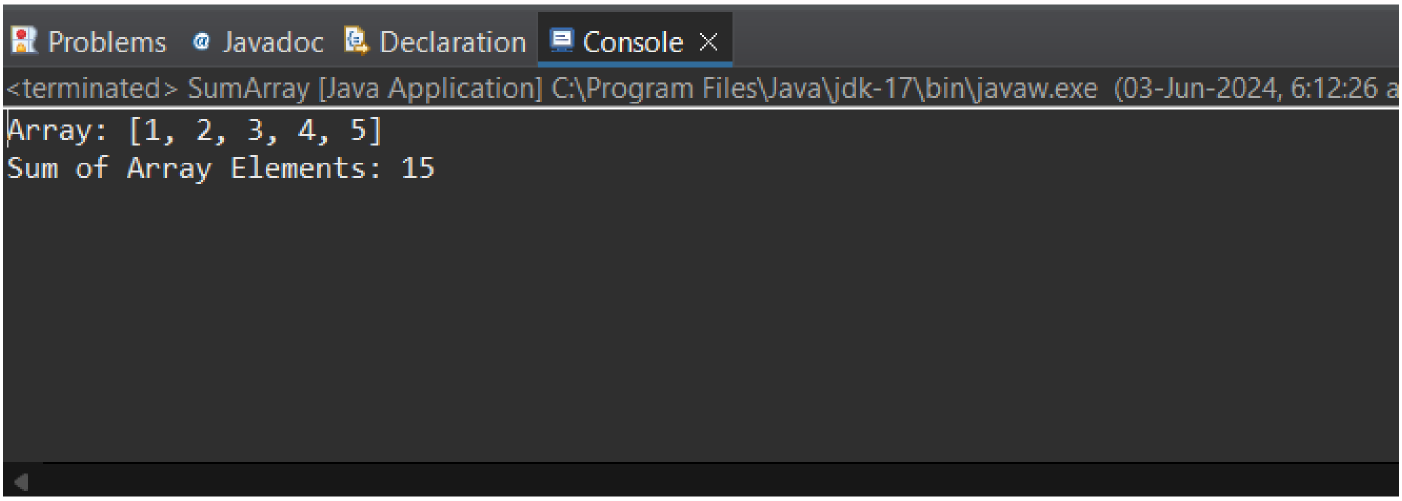
a) Write a recursive function named SumArray that calculates and returns the sum of elements in an array, demonstarte with example.

Code:



```java
package assignment.day_04;

import java.util.Arrays;

public class SumArray {
    public static void main(String[] args) {
        int[] arr = { 1, 2, 3, 4, 5 };
        System.out.println("Array: " + Arrays.toString(arr));
        int sum = SumArray(arr, arr.length);
        System.out.println("Sum of Array Elements: " + sum);
    }

    private static int SumArray(int[] arr, int n) {
        if (n <= 0)
            return 0;
        return SumArray(arr, n - 1) + arr[n - 1];
    }
}
```

Output:



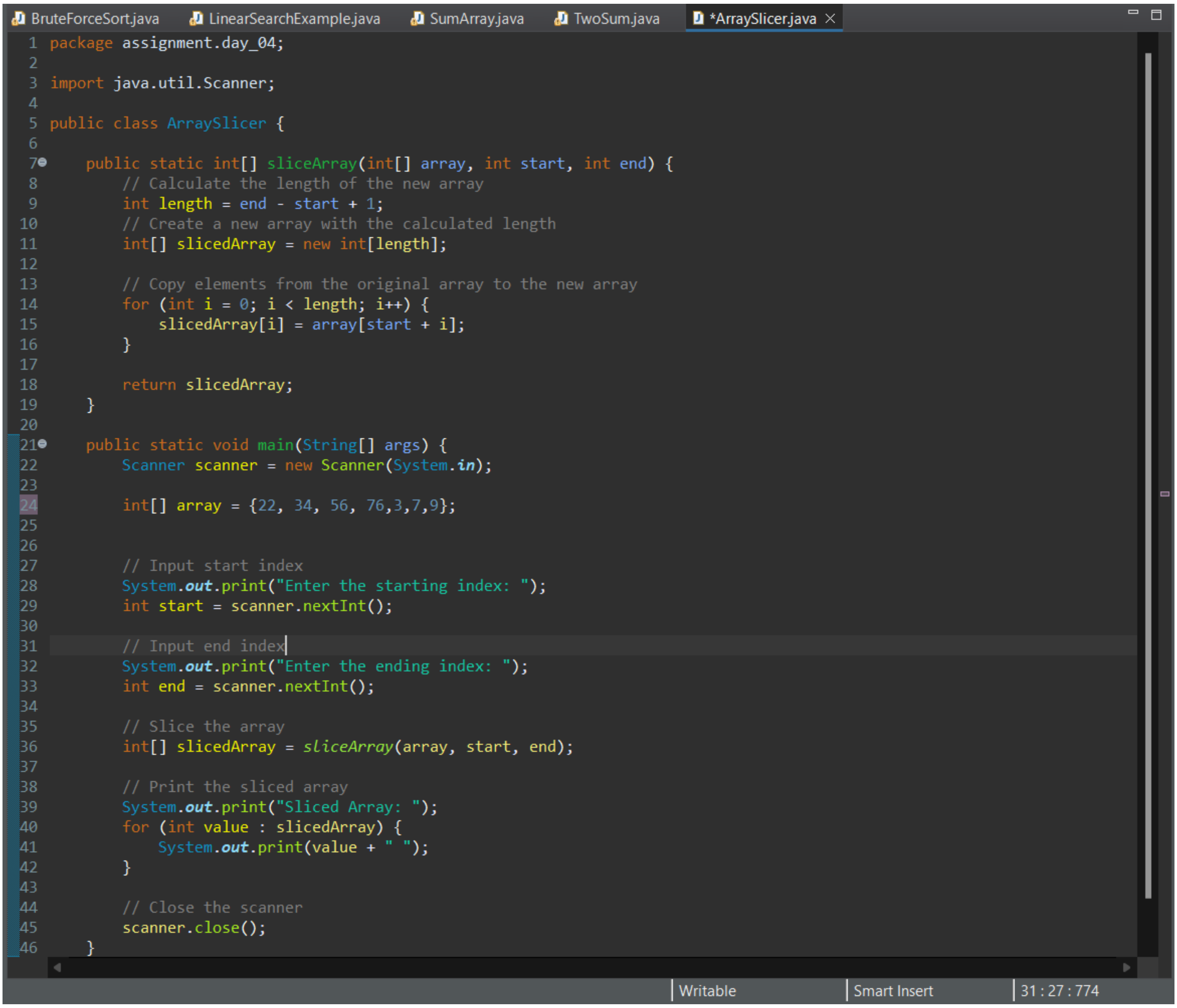
<terminated> SumArray [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (03-Jun-2024, 6:12:26 a

```
Array: [1, 2, 3, 4, 5]
Sum of Array Elements: 15
```

3Task 4: Advanced Array Operations

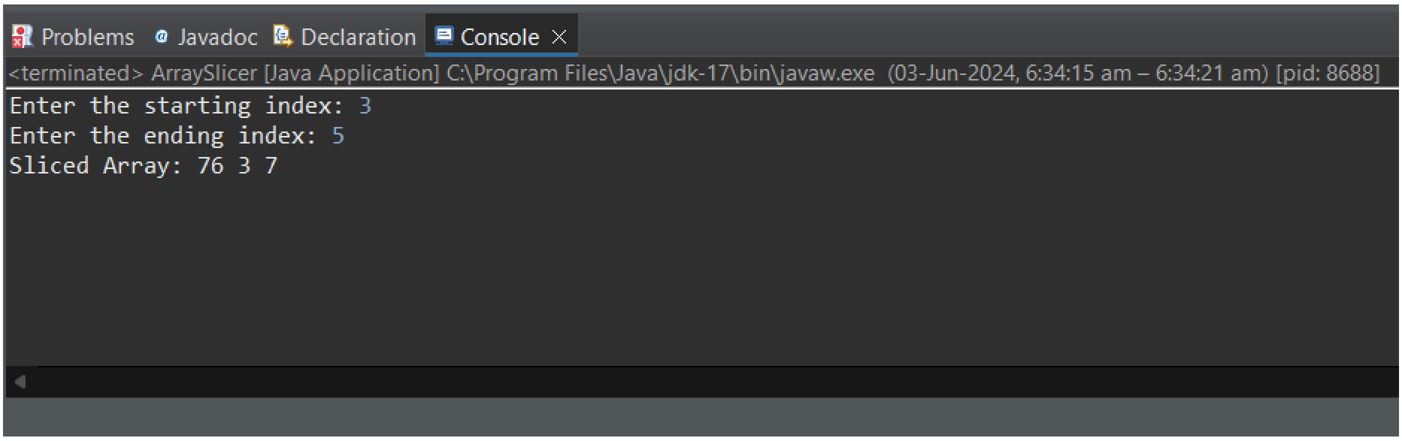
a) Implement a method SliceArray that takes an array, a starting index, and an end index, then returns a new array containing the elements from the start to the end index.

Code:



```java
package assignment.day_04;

import java.util.Scanner;

public class ArraySlicer {

    public static int[] sliceArray(int[] array, int start, int end) {
        // Calculate the length of the new array
        int length = end - start + 1;
        // Create a new array with the calculated length
        int[] slicedArray = new int[length];

        // Copy elements from the original array to the new array
        for (int i = 0; i < length; i++) {
            slicedArray[i] = array[start + i];
        }

        return slicedArray;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int[] array = {22, 34, 56, 76,3,7,9};


        // Input start index
        System.out.print("Enter the starting index: ");
        int start = scanner.nextInt();

        // Input end index
        System.out.print("Enter the ending index: ");
        int end = scanner.nextInt();

        // Slice the array
        int[] slicedArray = sliceArray(array, start, end);

        // Print the sliced array
        System.out.print("Sliced Array: ");
        for (int value : slicedArray) {
            System.out.print(value + " ");
        }

        // Close the scanner
        scanner.close();
    }
```

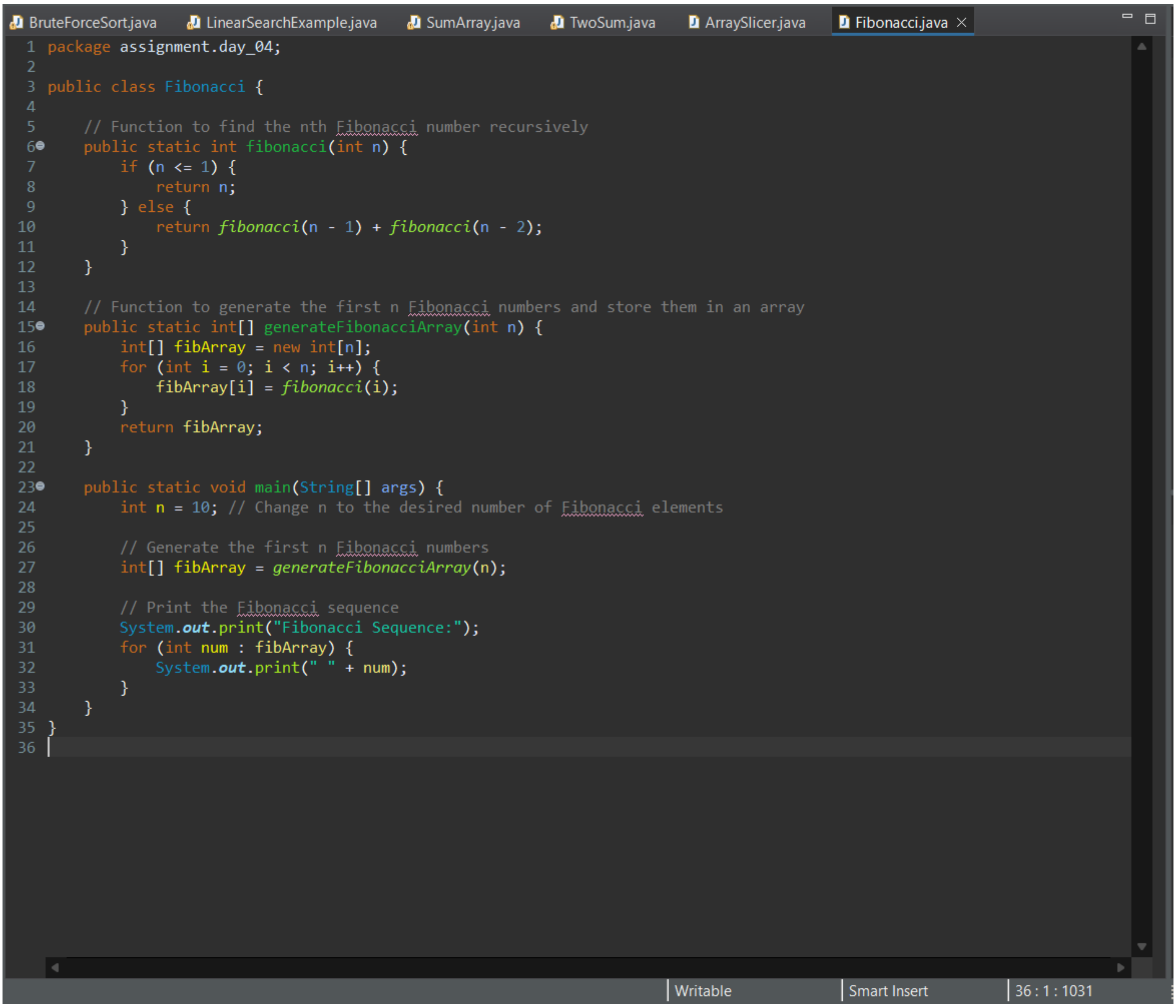
Output:



```
<terminated> ArraySlicer [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (03-Jun-2024, 6:34:15 am – 6:34:21 am) [pid: 8688]
Enter the starting index: 3
Enter the ending index: 5
Sliced Array: 76 3 7
```

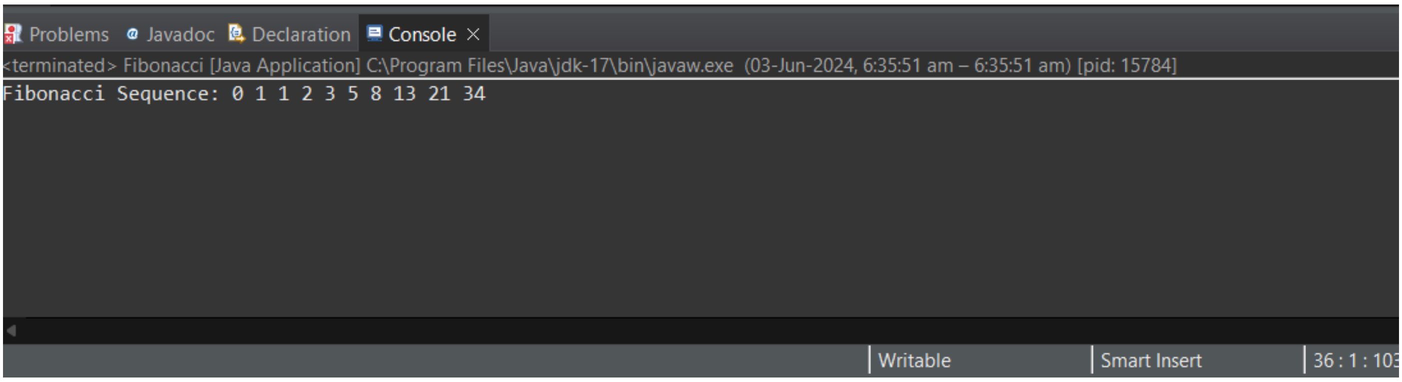
b) Create a recursive function to find the nth element of a Fibonacci sequence and store the first n elements in an array.

Code:



```java
package assignment.day_04;

public class Fibonacci {

    // Function to find the nth Fibonacci number recursively
    public static int fibonacci(int n) {
        if (n <= 1) {
            return n;
        } else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }

    // Function to generate the first n Fibonacci numbers and store them in an array
    public static int[] generateFibonacciArray(int n) {
        int[] fibArray = new int[n];
        for (int i = 0; i < n; i++) {
            fibArray[i] = fibonacci(i);
        }
        return fibArray;
    }

    public static void main(String[] args) {
        int n = 10; // Change n to the desired number of Fibonacci elements

        // Generate the first n Fibonacci numbers
        int[] fibArray = generateFibonacciArray(n);

        // Print the Fibonacci sequence
        System.out.print("Fibonacci Sequence:");
        for (int num : fibArray) {
            System.out.print(" " + num);
        }
    }
}
```

Output:



```
<terminated> Fibonacci [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (03-Jun-2024, 6:35:51 am – 6:35:51 am) [pid: 15784]
Fibonacci Sequence: 0 1 1 2 3 5 8 13 21 34
```