Credit
3110
Assignment
E2TernarySearch

How has your program changed from planning to coding to now? Please explain?

```
 9 School: CHHS
10 Course: Computer Programming 30
11 */
12
13 import java.util.Random;
14 import java.util.Scanner;
15
16 public class TernarySearchTest
17 {
18
19     public static void main(String[] args)
20     {
21         Scanner input = new Scanner(System.in);
22         Random rand = new Random();
23
24         // get array size
25         System.out.print("enter number of elements ");
26         int size = input.nextInt();
27
28         int[] numbers = new int[size];
29
30         // fill array
31         for (int i = 0; i < numbers.length; i++)
32         {
33             numbers[i] = rand.nextInt(90) + 10;
34         }
35
36         // display unsorted
37         System.out.print("unsorted: ");
38         displayArray(numbers);
39
40         // sort array
41         Sorts.mergeSort(numbers, 0, numbers.length - 1);
42
43         // display sorted
44         System.out.print("sorted:   ");
45         displayArray(numbers);
46
47         // get search value
48         System.out.print("enter number to search ");
49         int target = input.nextInt();
50
51         int index = Searches.ternarySearch(numbers, target, 0, numbers.length - 1);
52
53         // output result
```

```
        for (int i = 0; i < numbers.length; i++)
        {
            numbers[i] = rand.nextInt(90) + 10;
        }

        // display unsorted
        System.out.print("unsorted: ");
        displayArray(numbers);

        // sort array
        Sorts.mergeSort(numbers, 0, numbers.length - 1);

        // display sorted
        System.out.print("sorted:   ");
        displayArray(numbers);

        // get search value
        System.out.print("enter number to search ");
        int target = input.nextInt();

        int index = Searches.ternarySearch(numbers, target, 0, numbers.length - 1);

        // output result
        if (index == -1)
        {
            System.out.println("number not in list");
        }
        else
        {
            System.out.println("number found at index " + index);
        }

        input.close();
    }

    // displays array values
    private static void displayArray(int[] items)
    {
        for (int i = 0; i < items.length; i++)
        {
            System.out.print(items[i] + " ");
        }

        System.out.println();
    }
}
```

```java
Program: Searches.java
Last Date of this Revision: January 29, 2026

Purpose: provide search algorithms including ternary search
Author: Bilal Hajar
School: CHHS
Course: Computer Programming 30
*/

public class Searches
{

    /**
     * performs ternary search recursively
     * pre array sorted
     * post index or -1 returned
     */
    public static int ternarySearch(int[] items, int target, int left, int right)
    {
        if (left > right)
        {
            return -1;
        }

        int mid1 = left + (right - left) / 3;
        int mid2 = right - (right - left) / 3;

        // check mid points
        if (items[mid1] == target)
        {
            return mid1;
        }

        if (items[mid2] == target)
        {
            return mid2;
        }

        if (target < items[mid1])
        {
            return ternarySearch(items, target, left, mid1 - 1);
        }
        else if (target > items[mid2])
        {
            return ternarySearch(items, target, mid2 + 1, right);
```

```java

public class Searches
{

    /**
     * performs ternary search recursively
     * pre array sorted
     * post index or -1 returned
     */
    public static int ternarySearch(int[] items, int target, int left, int right)
    {
        if (left > right)
        {
            return -1;
        }

        int mid1 = left + (right - left) / 3;
        int mid2 = right - (right - left) / 3;

        // check mid points
        if (items[mid1] == target)
        {
            return mid1;
        }

        if (items[mid2] == target)
        {
            return mid2;
        }

        if (target < items[mid1])
        {
            return ternarySearch(items, target, left, mid1 - 1);
        }
        else if (target > items[mid2])
        {
            return ternarySearch(items, target, mid2 + 1, right);
        }
        else
        {
            return ternarySearch(items, target, mid1 + 1, mid2 - 1);
        }
    }
}
```