

**Assignment 3:** Write a program to implement binary search on given set of values and search key

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

        // Taking input for the array
        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.print("Enter the array elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        // Sorting the array using custom method
        bubbleSort(arr);

        // Sorting the array using array method
        Arrays.sort(arr);

        // Taking input for the element to search
        System.out.print("Enter the element to search: ");
        int target = scanner.nextInt();

        int result = binarySearch(arr, target);

        if (result == -1) {
            System.out.println("Element not found in the array.");
        } else {
            System.out.println("Element found at index " + result);
        }

        scanner.close();
    }

    // Bubble sort implementation
    static void bubbleSort(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    // Swap arr[j] and arr[j+1]
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
        System.out.println("sorted array : "+Arrays.toString(arr));
    }

    // Binary search implementation
    static int binarySearch(int[] arr, int target) {
        int low = 0;
        int high = arr.length - 1;

        while (low <= high) {
            int mid = low + (high - low) / 2;

            if (arr[mid] == target) {
                return mid;
            } else if (arr[mid] < target) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }

        return -1; // Element not found
    }
}
```

**Output:**

Enter the number of elements in the array: 5

Enter the array elements: 4 2 1 3 5

sorted array: [1, 2, 3, 4, 5]

Enter the element to search: 3

Element found at index 2