

Task . Search for a given element in a sorted array using Binary Search.

```
public class BinarySearchExample {  
    public static void main(String[] args) {  
        int[] numbers = {10, 20, 30, 40, 50, 60, 70};  
        int target = 40;  
  
        int result = binarySearch(numbers, target);  
  
        if (result == -1) {  
            System.out.println("Element not found.");  
        } else {  
            System.out.println("Element found at index: " + result);  
        }  
    }  
  
    public static int binarySearch(int[] arr, int target) {  
        int left = 0;  
        int right = arr.length - 1;  
  
        while (left <= right) {  
            int mid = left + (right - left) / 2;  
  
            if (arr[mid] == target) {  
                return mid; // found  
            }  
            if (arr[mid] < target) {  
                left = mid + 1; // search right half  
            } else {  
                right = mid - 1; // search left half  
            }  
        }  
        return -1; // not found  
    }  
}
```

```
        right = mid - 1; // search left half
    }
}

return -1; // not found
}

}
```

The screenshot shows a Java code editor interface with the following details:

- Code Area:** The file name is `BinarySearchExample.java`. The code implements a binary search algorithm. Lines 18-33 are shown, including the search loop, comparison logic, and base cases.
- Run Tab:** A blue "Run" button is visible at the top of the editor.
- Output Area:** To the right of the code, the output of the execution is displayed:
 - Element found at index: 3
 - ==== Code Execution Successful ===