

**DATA STRUCTURES &**

**ALGORITHMS**

**CS106.3**

**Coursework 2023/2024**

|  |  |
| --- | --- |
| **Name** | **J M S V JAYAWEERA** |
| **Index** | **29008** |
| **Degree** | **BSc (Hons) Software Engineering** |

**Coursework**

Use recursive approach to solve the Binary Search:

1. Write a proper pseudocode
2. Convert in to a program
3. Test the program with [2, 8, 9, 11, 15, 45, 58, 78, 99] array locating 2.

--------------------------------------- [Answer Below] ---------------------------------------

1. The Pseudocode

function binarySearch(arr, low, high, target):

if low <= high:

mid = (low + high) / 2

if arr[mid] == target:

return mid

else if arr[mid] < target:

return binarySearch(arr, mid + 1, high, target)

else:

return binarySearch(arr, low, mid - 1, target)

else:

return -1

1. I have written the program in both Java and C to understand and practice the concept.

* Java

public class RecursiveBinarySearch {

public static int binarySearch(int[] arr, int low, int high, int target) {

if (low <= high) {

int mid = (low + high) / 2;

if (arr[mid] == target) {

return mid;

} else if (arr[mid] < target) {

return binarySearch(arr, mid + 1, high, target);

} else {

return binarySearch(arr, low, mid - 1, target);

}

} else {

return -1;

}

}

* C

#include <stdio.h>

int binarySearch(int arr[], int low, int high, int target) {

if (low <= high) {

int mid = (low + high) / 2;

if (arr[mid] == target) {

return mid;

} else if (arr[mid] < target) {

return binarySearch(arr, mid + 1, high, target);

} else {

return binarySearch(arr, low, mid - 1, target);

}

} else {

return -1;

}

}

1. I Have Tried recursive method to Solve Binary Search using the Both Languages C and Java.

* **Java Code**

// This is a code that i wrote to solve Binary search using recursive method

public class RecursiveBinarySearch {

public static int binarySearch(int[] arr, int low, int high, int target) {

if (low <= high) {

int mid = (low + high) / 2;

if (arr[mid] == target) {

return mid;

} else if (arr[mid] < target) {

return binarySearch(arr, mid + 1, high, target);

} else {

return binarySearch(arr, low, mid - 1, target);

}

} else {

return -1;

}

}

public static void main(String[] args) {

int[] arr = {2, 8, 9, 11, 15, 45, 58, 78, 99};

int target = 2;

int result = binarySearch(arr, 0, arr.length - 1, target);

if (result != -1) {

System.out.println("Element " + target + " is found at index " + result + ".");

} else {

System.out.println("Element " + target + " is not present in the array.");

}

}

}

* **C Code**

// This is a code that i wrote to solve Binary search using recursive method

#include <stdio.h>

int binarySearch(int arr[], int low, int high, int target) {

if (low <= high) {

int mid = (low + high) / 2;

if (arr[mid] == target) {

return mid;

} else if (arr[mid] < target) {

return binarySearch(arr, mid + 1, high, target);

} else {

return binarySearch(arr, low, mid - 1, target);

}

} else {

return -1;

}

}

int main() {

int arr[] = {2, 8, 9, 11, 15, 45, 58, 78, 99};

int n = sizeof(arr) / sizeof(arr[0]);

int target = 2;

int result = binarySearch(arr, 0, n - 1, target);

if (result != -1) {

printf("Element %d is found at index %d.\n", target, result);

} else {

printf("Element %d is not present in the array.\n", target);

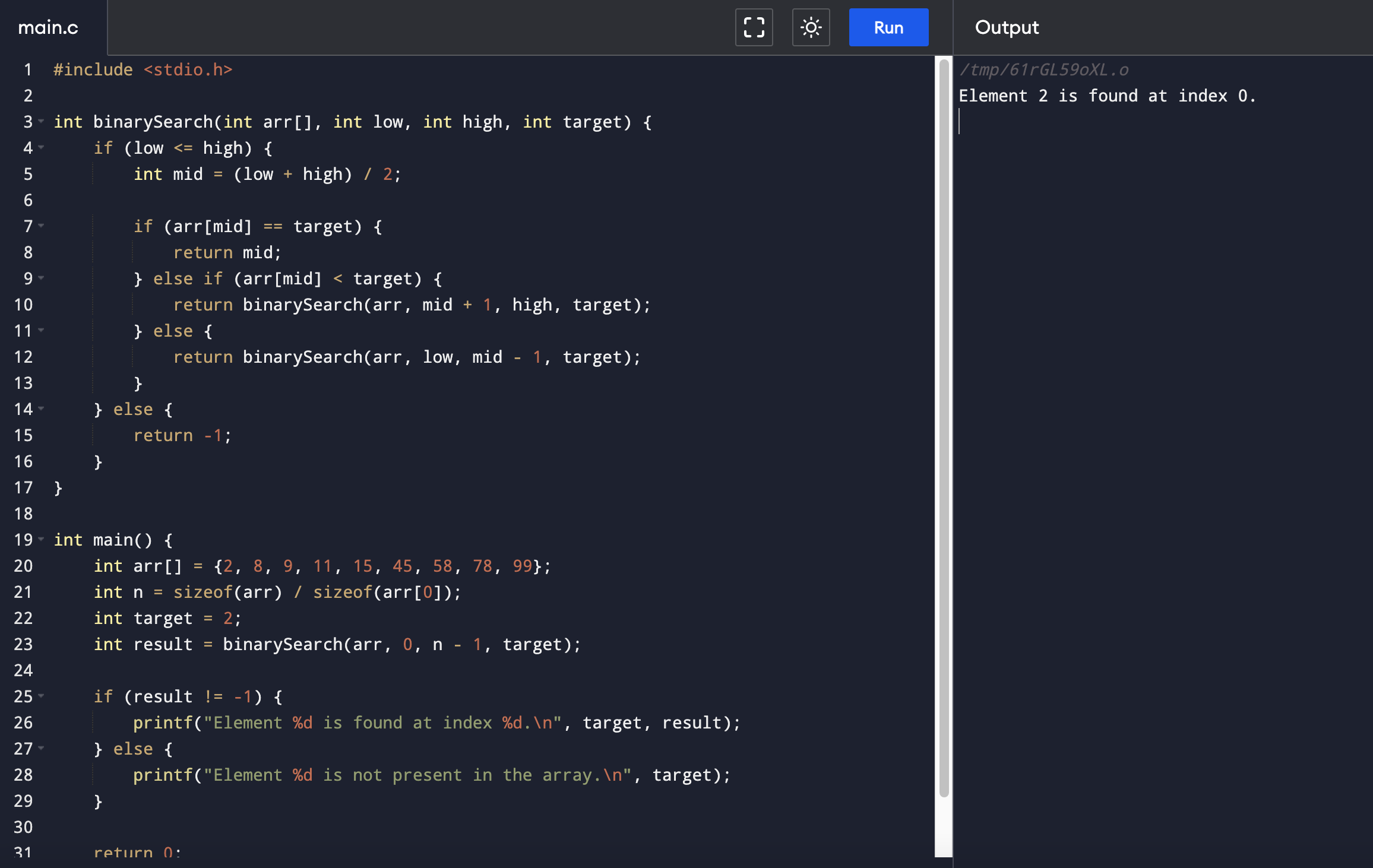
}

return 0;

}

* The Output

The output using C



The Output Using Java

