

Program for search in Rotated Sorted Array

```
class RotatedSortedArraySearch {

    public static int search(int[] arr, int target) {
        return binarySearch(arr, 0, arr.length - 1, target);
    }

    public static int binarySearch(int[] arr, int low, int high, int target) {
        if (low > high) {
            return -1; // Element not found
        }

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

        if (arr[mid] == target) {
            return mid; // Element found
        }

        // Check which part of the array is sorted
        if (arr[low] <= arr[mid]) { // Left half is sorted
            if (target >= arr[low] && target <= arr[mid]) {
                return binarySearch(arr, low, mid - 1, target);
            } else {
                return binarySearch(arr, mid + 1, high, target);
            }
        } else { // Right half is sorted
            if (target >= arr[mid] && target <= arr[high]) {
                return binarySearch(arr, mid + 1, high, target);
            } else {
                return binarySearch(arr, low, mid - 1, target);
            }
        }
    }

    public static void main(String[] args) {
        int[] rotatedArray = {0,1,4,5,6,7};
        int target = 4;
        int index = search(rotatedArray, target);
        System.out.println("Target element Index = "+index);
    }
}
```

OUTPUT

Target element Index = 2

2. Find First And Last Position Of Element In Sorted Array

```
public class FindFirstLastPosition {

    // Function to find the first and last position of a target element in a sorted array
    public static int[] searchRange(int[] nums, int target) {
        int[] result = {-1, -1};

        // Find the first occurrence of the target element
        int firstIndex = findFirstIndex(nums, target);
        if (firstIndex != -1) {
            result[0] = firstIndex;
            result[1] = findLastIndex(nums, target);
        }

        return result;
    }

    // Binary search to find the first occurrence of the target element
    public static int findFirstIndex(int[] nums, int target) {
        int low = 0;
        int high = nums.length - 1;
        int result = -1;

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

            if (nums[mid] >= target) {
                high = mid - 1;
                if (nums[mid] == target) {
                    result = mid;
                }
            } else {
                low = mid + 1;
            }
        }

        return result;
    }

    // Binary search to find the last occurrence of the target element
    public static int findLastIndex(int[] nums, int target) {
        int low = 0;
        int high = nums.length - 1;
        int result = -1;

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

            if (nums[mid] <= target) {
                low = mid + 1;
                if (nums[mid] == target) {
                    result = mid;
                }
            }
        }

        return result;
    }
}
```

```

        result = mid;
    }
    } else {
        high = mid - 1;
    }
}

return result;
}

public static void main(String[] args) {
    int[] nums = {5, 7, 2, 7, 8, 5, 2, 8, 11, 10};
    int target = 8;

    int[] result = searchRange(nums, target);

    System.out.println("First and Last Position = " + result[0] + ", " +
result[1] );
}
}

```

OUTPUT

First and Last Position = 4, 7