Task 8
Circular Queue Binary Search
Consider a circular queue (implemented using a fixed-size array) where the elements are sorted but have been rotated at an unknown index. Describe an approach to perform a binary search for a given element within this circular queue.

ANS:

```java
package Assigmentday12.com;
public class Task8 {
    public static int search(int[] array, int target) {
        int left = 0;
        int right = array.length - 1;
        while (left <= right) {
            int mid = left + (right - left) / 2;
            if (array[mid] == target) {
                return mid;
            }
            // Determine which half is sorted
            if (array[left] <= array[mid]) { // Left half is sorted
                if (target >= array[left] && target < array[mid]) {
                    right = mid - 1;
                } else {
                    left = mid + 1;
                }
            } else { // Right half is sorted
                if (target > array[mid] && target <= array[right]) {
                    left = mid + 1;
                } else {
                    right = mid - 1;
                }
            }
        }
        return -1; // Target not found
    }
    public static void main(String[] args) {
        int[] array = {12, 14, 18, 21, 3, 6, 8, 9}; // Circular sorted array
        int target = 6;
        int result = search(array, target);
        if (result != -1) {
            System.out.println("Element found at index: " + result);
        } else {
            System.out.println("Element not found");
        }
    }
}
```

OUTPUT:

```
Element found at index: 5
```