**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.**

**Sol:**

**Program:**

public class CircularQueueBinarySearch {

public int search(int[] nums, int target) {

int start = 0;

int end = nums.length - 1;

while (start <= end) {

int mid = start + (end - start) / 2;

if (nums[mid] == target) {

return mid;

}

// Determine which side is sorted

if (nums[start] <= nums[mid]) {

// Left side is sorted

if (nums[start] <= target && target < nums[mid]) {

end = mid - 1;

} else {

start = mid + 1;

}

} else {

// Right side is sorted

if (nums[mid] < target && target <= nums[end]) {

start = mid + 1;

} else {

end = mid - 1;

}

}

}

// Target is not found

return -1;

}

public static void main(String[] args) {

CircularQueueBinarySearch searcher = new CircularQueueBinarySearch();

int[] nums = {4, 5, 6, 7, 0, 1, 2};

int target = 0;

int index = searcher.search(nums, target);

System.out.println("Index of " + target + " is: " + index);

}

}