**Finding Zero Sum Subarray (Kadane's Algorithm Variant):**

function subArrayZeroKadane(nums) {

  //TC: O(n)

  let sum = 0;

  let startIndex = 0;

  let endIndex = 0;

  for (let i = 0; i < nums.length; i++) {

    endIndex++;

    sum += nums[i];

    if (sum === 0) {

      break;

    }

    if (sum < 0) {

      sum = nums[i];

      startIndex = i;

    }

  }

  return {

    zeroSubArray: nums.slice(startIndex, endIndex),

  };

}

console.log(subArrayZeroKadane([2, 1, -4, 4, -1, 2, 1, -5, 4]));

**Binary Search in Sorted Array:**

const arr = [2, 4, 5, 7, 9, 12]

function findMe(target, start, end){

    if(start>end){

        return "Not Found"

    }

    const middle = Math.floor((start+end)/2)

    if(arr[middle]==target){

        return `Found it at index ${middle}`

    }

    if(arr[middle]>target){

        return findMe(target, start, middle-1)

    }

    if(arr[middle]<target){

        return findMe(target,  middle+1, end)

    }

}

console.log(findMe(7, 0, 5))