Lab Assignment: Binary Search in a Sorted Array (Recursive)

Problem Statement

Given a sorted array of integers (in ascending order), determine whether a given number (key) is present in the array.

- Use the recursive version of Binary Search.
- The array may contain duplicate elements.
- Return the index of the key if found; otherwise, return -1.

Function Prototype

```
int binarySearch(int arr[], int low, int high, int key);
```

Binary Search Logic (Recursive Version)

1. Start with two indices: low (start of the array) and high (end of the array).
2. Base Case: If low > high, return -1 (key not found).
3. Calculate the Midpoint:
   mid = (low + high) / 2
4. Compare the key with the middle element:
   - If arr[mid] == key, return mid.
   - If key < arr[mid], recursively search the left subarray:
     binarySearch(arr, low, mid - 1, key)
   - If key > arr[mid], recursively search the right subarray:
     binarySearch(arr, mid + 1, high, key)
5. Continue the recursive calls until the base case is reached.

Example Cases

Example 1:

Input: arr = [1, 3, 5, 7, 9, 11, 13], key = 7

Output: Index of 7: 3

Explanation: The number 7 is found at index 3 (0-based index).

Lab Assignment: Binary Search in a Sorted Array (Recursive)

Example 2:

Input: arr = [2, 4, 6, 8, 10], key = 5

Output: Index of 5: -1

Explanation: 5 is not present in the array.