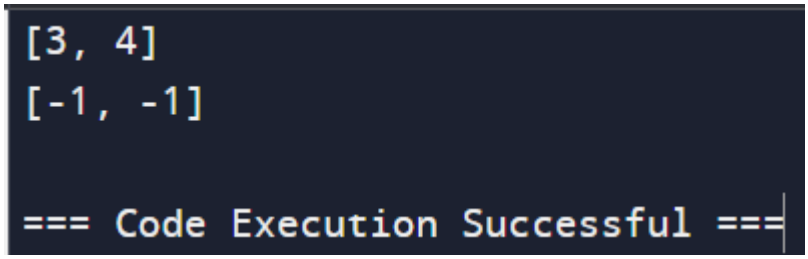
45. Find First and Last Position of Element in Sorted Array Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value. If target is not found in the array, return [-1, -1]. You must write an algorithm with O(log n) runtime complexity. Example 1: Input: nums = [5,7,7,8,8,10], target = 8 Output: [3,4

Program:-

```
def findFirstAndLast(nums, target):
    def binarySearchLeft(nums, target):
        left, right = 0, len(nums) - 1
        while left <= right:
            mid = (left + right) // 2
            if nums[mid] < target:
                left = mid + 1
            else:
                right = mid - 1
        return left

    def binarySearchRight(nums, target):
        left, right = 0, len(nums) - 1
        while left <= right:
            mid = (left + right) // 2
            if nums[mid] <= target:
                left = mid + 1
            else:
                right = mid - 1
        return right

    start = binarySearchLeft(nums, target)
    end = binarySearchRight(nums, target)

    if start <= end and 0 <= start < len(nums) and nums[start] == target:
        return [start, end]
    else:
        return [-1, -1]

# Example usage:
nums = [5,7,7,8,8,10]
target = 8
result = findFirstAndLast(nums, target)
print(result)  # Output: [3, 4]

# Another example:
nums = [5,7,7,8,8,10]
target = 6
result = findFirstAndLast(nums, target)
print(result)  # Output: [-1, -1]
```

output:-



TIME COMPLEXITY:-O(log n)