Program 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.

PROGRAM:

```
def searchRange(nums, target):
    def findBound(isFirst):
        left, right = 0, len(nums) - 1
        while left <= right:
            mid = (left + right) // 2
            if nums[mid] > target or (isFirst and nums[mid] == target):
                right = mid - 1
            else:
                left = mid + 1
        return left

    first = findBound(True)
    last = findBound(False) - 1

    if first <= last < len(nums) and nums[first] == target:
        return [first, last]
    return [-1, -1]

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

Output::

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 45.py"
[3, 4]

Process finished with exit code 0
```

Time complexity:
O(log n)