**6.A peak element is an element that is strictly greater than its neighbors. Given a 0-indexed integer array nums, find a peak element, and return its index. If the array contains multiple peaks, return the index to any of the peaks. You may imagine that nums[-1] = nums[n] = -∞. In other words, an element is always considered to be strictly greater than a neighbor that is outside the array. You must write an algorithm that runs in O(log n) time.**

**Example 1:**

**Input: nums = [1,2,3,1]**

**Output: 2**

**Explanation: 3 is a peak element and your function should return the index number 2.**

**Example 2:**

**Input: nums = [1,2,1,3,5,6,4]**

**Output: 5**

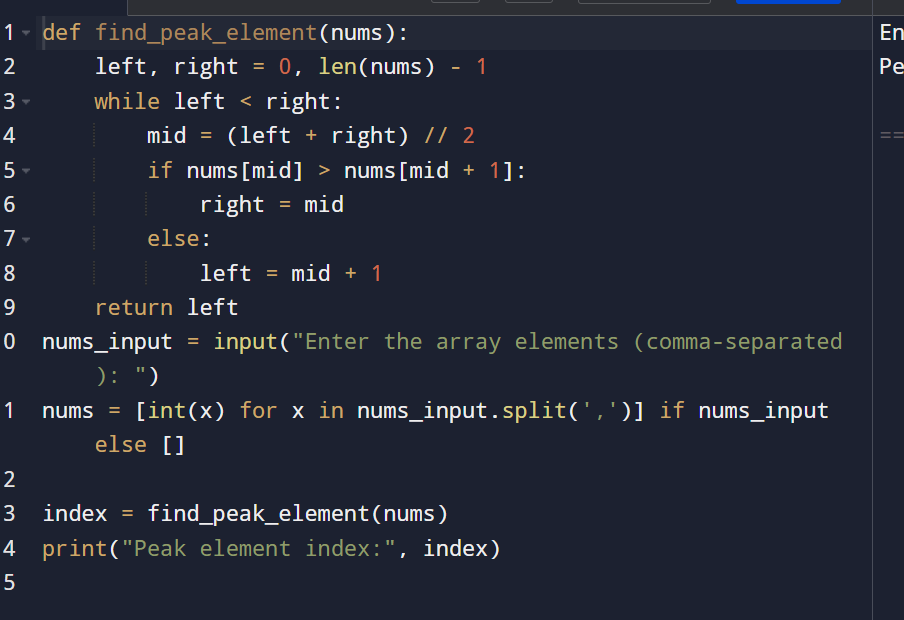
**Aim:**

To find the index of any peak element (strictly greater than its neighbors) in an array using an efficient O(log n) binary search algorithm.

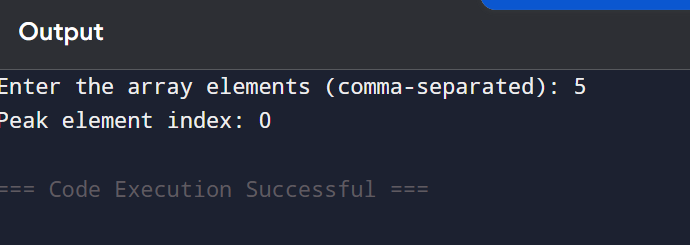
**Algorithm:**

1. Use binary search on the array.
2. At each step, calculate mid index.
3. Compare nums[mid] with nums[mid + 1]:
   * If nums[mid] > nums[mid + 1], a peak exists on the left side (including mid).
   * Else, a peak exists on the right side.
4. Narrow the search until left == right; return left (or right) as the peak index.

**Code:**

****

**Input and output:**

****

**Result: the program is executed successfully and output is verified**

**Performance analysis:**

**Time Complexity: O(log n)**

**Space Complexity: O(1)**