**AP CSA FRQ Practice**

**Question 1:**
Write a method longestIncreasingSequence that takes an int[] array as a parameter and returns the length of the longest consecutive increasing sequence in the array.
*Example:* If the array is {2, 3, 5, 1, 2, 3, 4}, the method should return 4 (for the sequence 1, 2, 3, 4).

---

**Question 2:**
Write a method shiftRight that takes an ArrayList<Integer> and an integer n as parameters and shifts all elements to the right by n positions. Elements shifted past the last index should wrap around to the front of the list.
*Example:* If the list is [1, 2, 3, 4, 5] and n = 2, after calling shiftRight, the list should be [4, 5, 1, 2, 3].

---

**Question 3:**
Write a method hasAdjacentDuplicates that takes an int[] array as a parameter and returns true if the array contains **any two identical consecutive elements**, and false otherwise.
*Example:* For the array {1, 3, 3, 7}, the method should return true. For {1, 2, 3, 4}, it should return false.

---

**Question 4:**
Write a method transpose that takes a square int[][] array as a parameter and returns a new 2D array that is the transpose of the original. The transpose of a matrix flips rows and columns (i.e., the element at [i][j] becomes [j][i]).

---

**Question 5:**
Write a method countLocalMaxima that takes a 2D int array and returns the number of elements that are **strictly greater than all of their immediate neighbors** (top, bottom, left, right, and diagonal). Assume the array has at least 1 row and 1 column.

---

**Question 6:**

Write a method replaceNegatives that takes an int[][] array and replaces every negative value with the sum of its immediate neighbors (top, bottom, left, right). If a neighbor does not exist (edge or corner), it is ignored.