**Program - 1:**

Create a program that iterates through an array 'n' times. During each iteration, move the last element to the first position of the array and shift all other elements to the right by one index.

For example, given the input array [c,o,d,i,n,g,h,a,p,p,y] and n=5, the output array would be [h,a,p,p,y,c,o,d,i,n,g].

**Program - 2:**

Identify duplicates in a two-dimensional integer array and increment each duplicate element by 1. If the incremented value also creates a duplicate, then continue incrementing by 1 until a unique value is obtained.

**Sample Input:**

[[3,2,3,2], [1,2,3,3], [3,2,1,3]]

**Sample Output:**

[[3,2,4,5], [1,2,3,4], [3,2,1,4]]

**Program - 3:**

How do you print an inverted pyramid of asterisks (\*) using programming, where the number of asterisks in the top row is the highest, and the number of asterisks decreases as we move down the rows, forming an inverted pyramid shape with a specified number of rows?

**Program - 4:**

Write a program that takes in a string of words and outputs the frequency of each word in the string, sorted in descending order of frequency. For instance, if the input string is "the quick brown fox jumps over the lazy dog", the output should be:

the: 2

over: 1

lazy: 1

jumps: 1

fox: 1

brown: 1

quick: 1

dog: 1

**Program 5:**

Write a program that takes in two arrays of integers and returns an array containing the union of the two arrays (i.e., all distinct elements from both arrays), sorted in ascending order. The program should have a time complexity of O(n log n) or better. For instance, if the input arrays are [2, 5, 1, 7] and [3, 6, 5, 8], the output should be [1, 2, 3, 5, 6, 7, 8].