**THE FULL COUNTING SORT**

Counting Sort is a sorting algorithm that works by counting the occurrences of each element in the input and then arranging them in order based on their counts. It is particularly useful when the range of input numbers is limited. In this implementation, we are sorting a 2D array where each element has a string and a number, similar to a scenario where we have labeled data that needs ordering by numerical value.

The algorithm starts by initializing an array of empty lists, called count, where the index represents the number from the input. For each element in the input, we determine its key, which is the numeric part. For the first half of the input, the string is replaced with a dash "-" as a placeholder, while for the second half, the original string is preserved. Each element is appended to its corresponding list in the count array based on its numeric key.

Once all elements are placed into the count array, we iterate through the count array sequentially to concatenate the lists into a single result. This produces the sorted order while maintaining stability, meaning elements with the same numeric key appear in the order they were initially read.

This approach efficiently sorts the input in **O(n + k)** time, where n is the number of elements and k is the maximum key value, making it faster than traditional comparison-based algorithms for suitable datasets.

**Implementation by:**  
Mark Anthony Alava  
BSIT-3 Nexus