LAB 2 EXERCISE

Task#1: you are required to declare an array and initialize it at run time then count the even and odd numbers in array. After that separate the even and odd arrays finally print all three arrays even array, odd array and complete array which contains even and odd numbers.

Task#2: Declare two arrays and initialize them at run time after that find the common elements from both. And make separate array for common elements.

Task#3: Write a program to calculate the GPA of students of all subjects of a single semester. Assume all the courses have the same credit hour (let's assume 3 credit hours). Do this task with Jagged Array.

Sample Data:

Names	**Data Structure**	**Digital Logic Design**	**OOP**	**Linear Algebra**	**Entrepreneurshi**p
Ali	3.21	2.45	4.0	3.0	3.75
Ahmed	3.75	2.0	--	2.15	--
Naseer	3.45	3.0	--	--	--
Zameer	4.0	3.0			
Waqar	3.33	2.75	3.35	3.98	

Task#4: Write a java program to ask the user enter the elements in unsorted form then sort the elements. If first sorted element is even then sort all even elements first then Odd elements and vice versa. Use **Bubble Sort** to do this.

Example 1: Unsorted array: 10, 1, 3, 4, 2, 5, 7, 6, 11

1. Sorted array: 1, 3, 5, 7, 11, 2, 4, 6, 10

Example 2: Unsorted array: 5, 4, 2, 7, 3, 8

1. Sorted array: 2, 4, 8, 3, 5, 7

Task#5: Modify the bubble sort implementation to sort the above array in descending order instead of ascending order. Ensure that the modified algorithm works correctly and compare its performance with the original implementation.

Task#6: Implement an optimized version of the bubble sort algorithm in above given array that stops early if no swaps are made in a pass.

Task#7: Write a function that implements the bubble sort algorithm to sort an array of strings in lexicographic (alphabetical) order. Test your implementation with an array of strings to ensure correct sorting.

Original Array: zebra apple banana grape date cherry
Sorted Array: apple banana cherry date grape zebra