**LAB # 04**

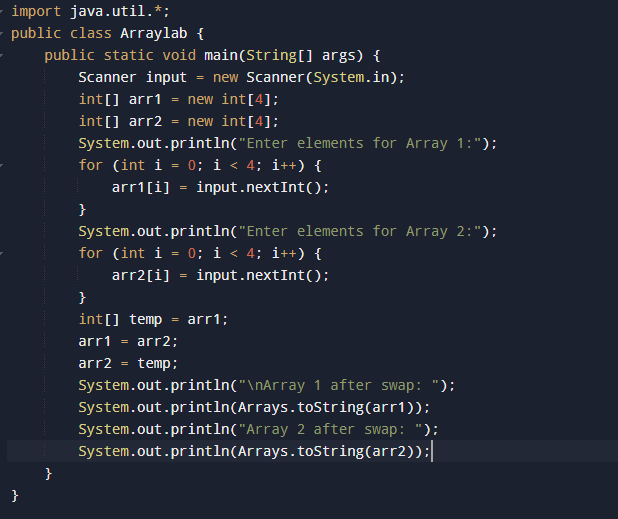
**ARRAYS IN JAVA**

**OBJECTIVE:** **To understand arrays and its memory allocation.**

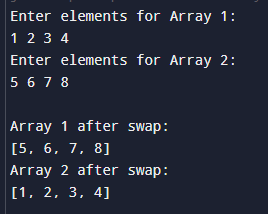
**LAB TASKS**

1. Write a program that takes two arrays of size 4 and swap the elements of those arrays

Lab task#1:

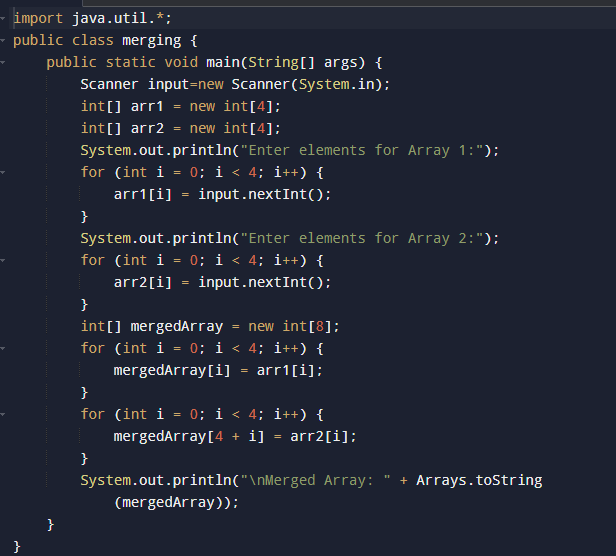


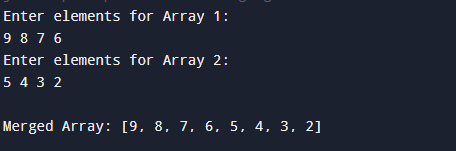
Output:



task#2:

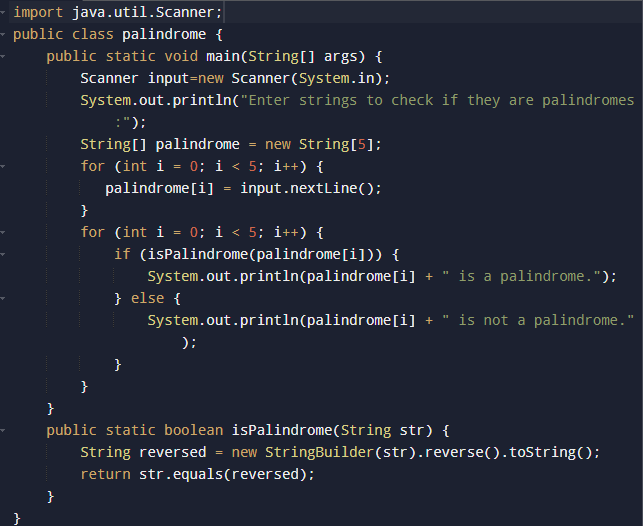
2. Add a method in the class that takes array and merge it with the existing one.

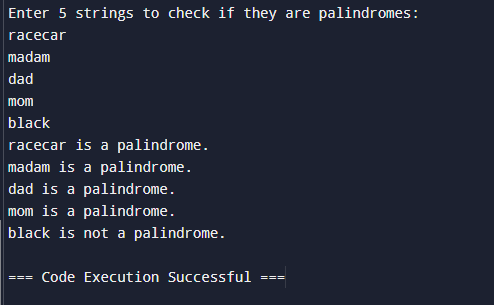


Output:

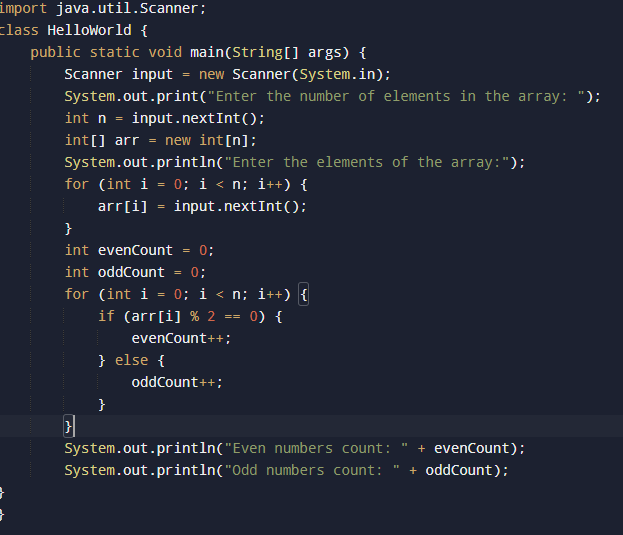
Lab task#3:

3. In a JAVA program, take an array of type string and then check whether the strings are palindrome or not.

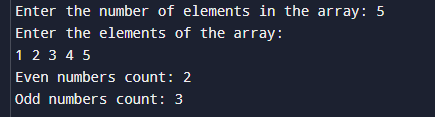


Output:

Lab task#4:

4. Given an array of integers, count how many numbers are even and how many are odd.

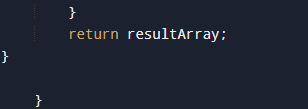
Output:

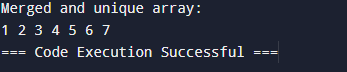


Lab task#5:

5. Given two integer arrays, merge them and remove any duplicate values from the resulting array.



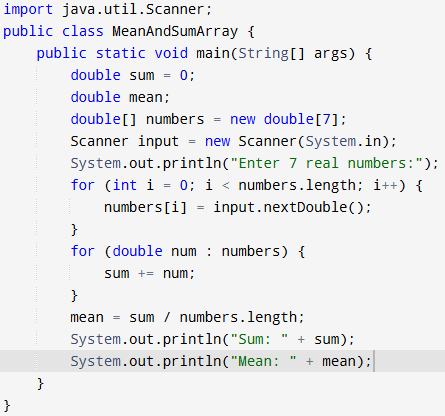


Output:

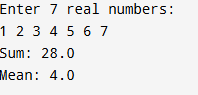
**Home tasks:**

Task#1:

1. Write a program that takes an array of Real numbers having size 7 and calculate the sum and mean of all the elements. Also depict the memory management of this task.

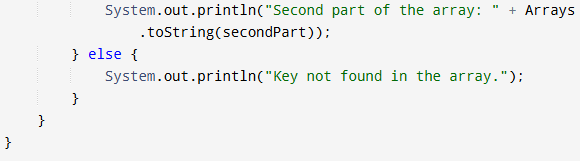
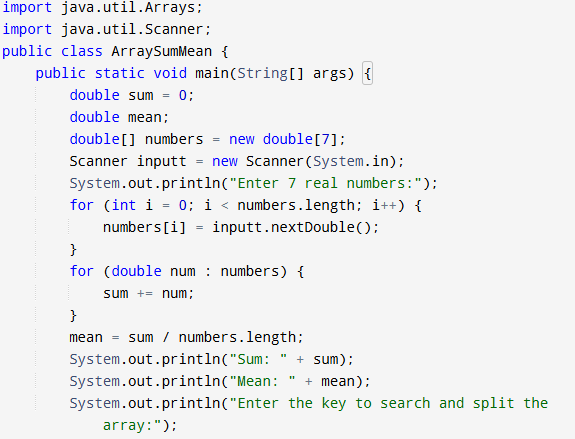
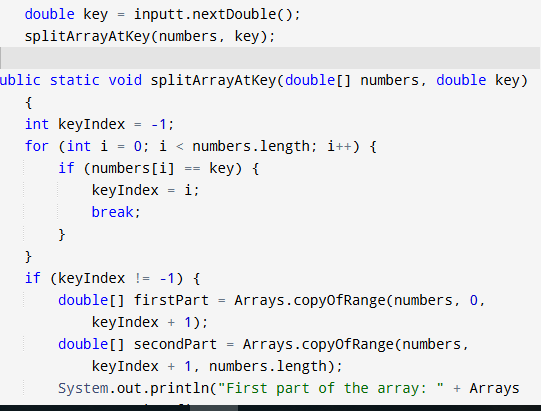


Output:

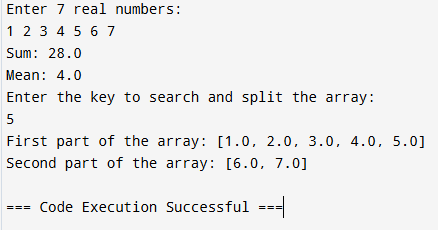


Task#2:

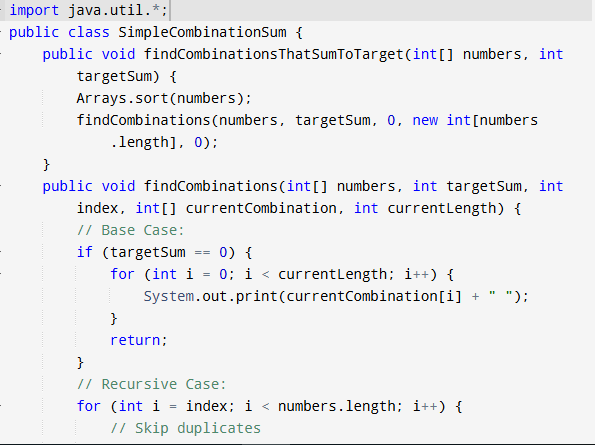
2. Add a method in the same class that splits the existing array into two. The method should search a key in array and if found splits the array from that index of the key

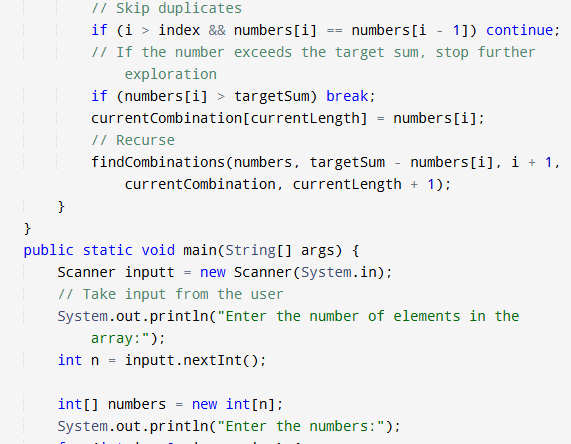


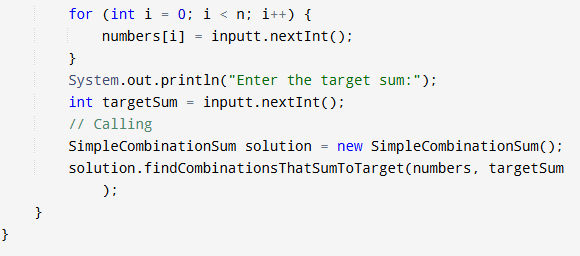
Output:

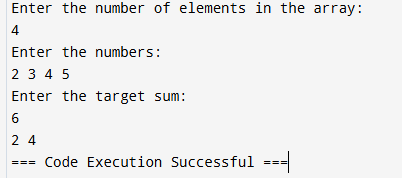


Task#3:

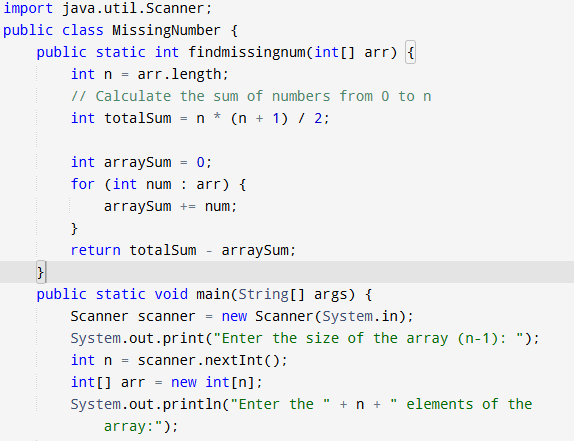
3. Given an array of distinct integers and a target integer, return all unique combinations of numbers that add up to the target. Each number can be used only once in the combination.

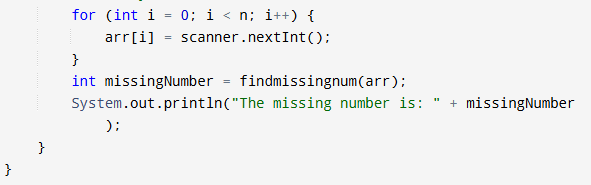


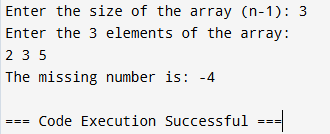


Output:

Task#4:

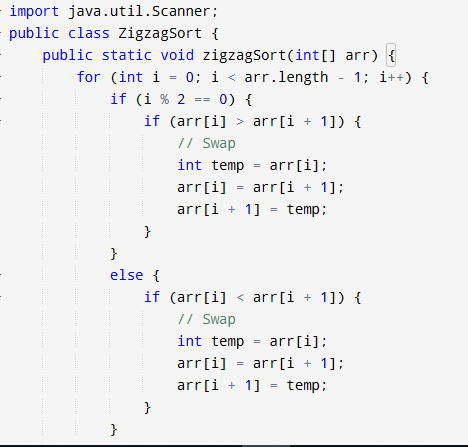
4. You are given an array containing n distinct numbers taken from 0, 1, 2, ..., n. Write a program to find the one number that is missing from the array

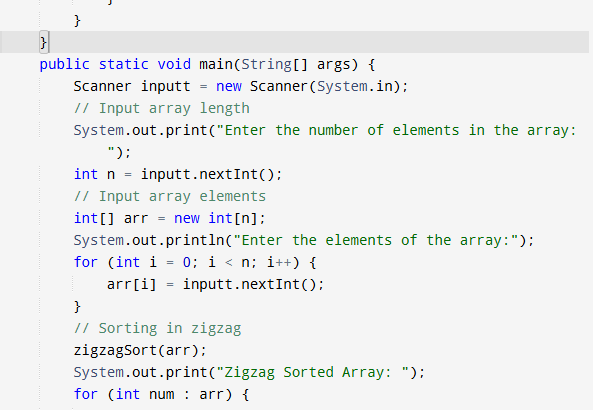


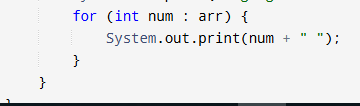
output:

Task#5:

5. You are given an array of integers. Write a program to sort the array such that it follows a zigzag pattern: the first element is less than the second, the second is greater than the third, and so on.







Output:

