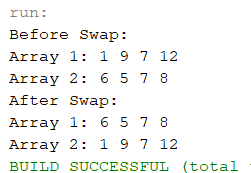
**LAB#04**

**Lab Task#01:**

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

**Code: Output:**

public class DSALab04 {

 public static void Print(int[] array1, int[] array2) {

System.out.print("Array 1: ");

for (int no : array1) System.out.print(no + " ");

System.out.print("\nArray 2: ");

for (int no : array2) System.out.print(no + " ");

System.out.println(); }

public static void main(String[] args) {

int[] ar1 = {1, 9, 7, 12};

int[] ar2 = {6, 5, 7, 8};

System.out.println("Before Swap:");

Print(ar1, ar2);

for (int i = 0; i < 4; i++) {

int temp = ar1[i];

ar1[i] = ar2[i];

ar2[i] = temp; }

System.out.println("After Swap:");

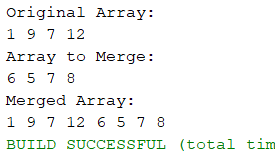
Print(ar1, ar2); } }

**Lab Task#02:**

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

**Code: Output:**

public class DSALab04 {

 public static void main(String[] args) {

int[] ar1 = {1, 9, 7, 12};

int[] ar2 = {6, 5, 7, 8};

System.out.println("Original Array:");

Print(ar1);

System.out.println("Array to Merge:");

Print(ar2);

int[] mergedArray = Merge(ar1, ar2);

System.out.println("Merged Array:");

Print(mergedArray); }

public static int[] Merge(int[] arr1, int[] arr2) {

int[] merged = new int[arr1.length + arr2.length];

for (int i = 0; i < arr1.length; i++) {

merged[i] = arr1[i]; }

for (int i = 0; i < arr2.length; i++) {

merged[arr1.length + i] = arr2[i]; }

return merged; }

public static void Print(int[] arr) {

for (int no : arr) {

System.out.print(no + " "); }

System.out.println(); }

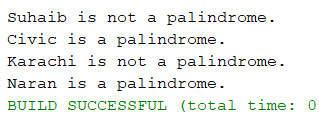
}

**Lab Task#03:**

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

**Code: Output:**

public class DSALab04 {

 public static boolean isPalindrome(String str) {

str = str.toLowerCase();

int left = 0, right = str.length() - 1;

while (left < right) {

if (str.charAt(left) != str.charAt(right)) {

return false; }

left++;

right--; }

return true; }

public static void main(String[] args) {

String[] words = {"Suhaib", "Civic", "Karachi", "Naran"};

for (String s : words) {

if (isPalindrome(s)) {

System.out.println(s + " is a palindrome.");

} else {

System.out.println(s + " is not a palindrome."); } }

}

}

**Lab Task#04:**

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

**Code: Output:**

public class DSALab04 {

 public static void main(String[] args) {

int[] no = {1, 2, 5, 6, 8, 9, 11, 12, 15, 16, 20, 21};

int even = 0, odd = 0;

for (int num : no) {

if (num % 2 == 0) even++;

else odd++; }

System.out.println("Even: " + even + ", Odd: " + odd);

}

}

**Lab Task#05:**

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

**Code: Output:**

import java.util.\*;

 public class DSALab04 {

public static int[] Duplicates(int[] arr1, int[] arr2) {

Set<Integer> set = new HashSet<>();

for (int no : arr1) set.add(no);

for (int no : arr2) set.add(no);

int[] result = new int[set.size()];

int i = 0;

for (int num : set) result[i++] = num;

return result; }

public static void main(String[] args) {

int[] arr1 = {4, 8, 6, 9, 11};

int[] arr2 = {6, 5, 2, 4, 9};

int[] mergedArray = Duplicates(arr1, arr2);

System.out.println("Updapted Array : " + Arrays.toString(mergedArray)); }

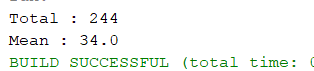
}

**Home Task#01:**

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.

**Code: Output:**

public class DSALab04 {

 public static void main(String[] args) {

int[] no = {11, 45, 3, 78, 36, 63, 8};

int total = 0;

for (int num : no) total += num;

double mean = total / 7;

System.out.println("Total : " + total);

System.out.println("Mean : " + mean); }

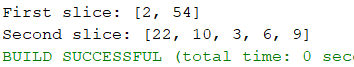
}

**Home Task#02:**

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.

**Code: Output:**

import java.util.\*;

 public class DSALab04 {

public static void Split(int[] arr, int key) {

int index = -1;

for (int i = 0; i < arr.length; i++) {

if (arr[i] == key) {

index = i;

break; } }

if (index != -1) {

int[] firstPart = Arrays.copyOfRange(arr, 0, index);

int[] secondPart = Arrays.copyOfRange(arr, index, arr.length);

System.out.println("First slice: " + Arrays.toString(firstPart));

System.out.println("Second slice: " + Arrays.toString(secondPart));

} else {

System.out.println("Key not found."); }

}

public static void main(String[] args) {

int[] no = {2, 54, 22, 10, 3, 6, 9};

int key = 22;

Split(no, key); }

}

**Home Task#03:**

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

**Code: Output:**

import java.util.\*;

public class DSALab04 {

public static List<List<Integer>> Combinations(int[] numbers, int target) {

List<List<Integer>> combinations = new ArrayList<>();

Arrays.sort(numbers);

findCombinationsHelper(combinations, new ArrayList<>(), numbers, target, 0);

return combinations; }

static void findCombinationsHelper(List<List<Integer>> combinations, List<Integer> current, int[] numbers, int target, int start) {

if (target == 0) {

combinations.add(new ArrayList<>(current));

return;}

for (int i = start; i < numbers.length && numbers[i] <= target; i++) {

if (i > start && numbers[i] == numbers[i - 1]) continue;

current.add(numbers[i]);

findCombinationsHelper(combinations, current, numbers, target - numbers[i], i + 1);

current.remove(current.size() - 1);}

}

public static void main(String[] args) {

int[] numbers = {10, 1, 2, 7, 6, 5};

int target = 8;

System.out.println("Combinations: " + Combinations(numbers, target));}

}



**Home Task#04:**

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.

**Code: Output:**

 public class DSALab04 {

public static int findMissing(int[] numbers) {

int n = numbers.length, sum = n \* (n + 1) / 2;

for (int num : numbers) sum -= num;

return sum;

}

public static void main(String[] args) {

int[] no = {3, 0, 1};

System.out.println("Missing number: " + findMissing(no));

}

}

**Home Task#05:**

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.

**Code: Output:**

import java.util.\*;

public class DSALab04 {

public static void zigzag(int[] arr) {

for (int i = 0; i < arr.length - 1; i++) {

if ((i % 2 == 0 && arr[i] > arr[i + 1]) || (i % 2 != 0 && arr[i] < arr[i + 1])) {

int temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp; }

}

}

public static void main(String[] args) {

int[] arr = {4, 3, 7, 6, 9, 2, 0};

zigzag(arr);

System.out.println(Arrays.toString(arr)); }

}