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.

CODE:

```
package lab4.pkg015;
import java.util.Arrays;
public class Lab4015 {
    public static void main(String[] args) {
        //task 1
        int[] array1 = {34,56,67,89};
        int[] array2 = {34,32,33,266};

        System.out.println("Before Swapping the values:");
        System.out.println("Array 1: " + Arrays.toString(array1));
        System.out.println("Array 2: " + Arrays.toString(array2));
        for (int i = 0; i < 4; i++) {
            int temp = array1[i];
            array1[i] = array2[i];
            array2[i] = temp;
        }
        System.out.println("\nAfter Swap:");
        System.out.println("Array 1: " + java.util.Arrays.toString(array1));
        System.out.println("Array 2: " + java.util.Arrays.toString(array2));
        System.out.println();
    }
}
```

OUTPUT:

```
run:
Before Swapping the values:
Array 1: [34, 56, 67, 89]
Array 2: [34, 32, 33, 266]

After Swap:
Array 1: [34, 32, 33, 266]
Array 2: [34, 56, 67, 89]

BUILD SUCCESSFUL (total time: 2 seconds)
```

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

CODE:

```
package lab4.pkg015;
import java.util.Arrays;
public class Lab4015 {
    public static void main(String[] args) {
        int[] arr15 = {4,277,44};
        int[] arr19 = {19,15,392};
        int[] mergedArray = mergeArrays(arr15, arr19);
        System.out.println(Arrays.toString(mergedArray));
    }
    public static int[] mergeArrays(int[] arr15, int[] arr19) {
        int[] mergedArray = new int[arr15.length + arr19.length];
        System.arraycopy(arr15, 0, mergedArray, 0, arr15.length);
        System.arraycopy(arr19, 0, mergedArray, arr15.length, arr19.length);
        return mergedArray;
    }
}
```

OUTPUT:

```
run:
[4, 277, 44, 19, 15, 392]
BUILD SUCCESSFUL (total time: 1 second)
```

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

CODE:

```
public class Lab4015 {
    public static void main(String[] args) {
         String[] lab4 = {"laiba", "sara", "racecar", "MOM"};
        for (String word : lab4) {
            if (isPalindrome(word)) {
                System.out.println(word + " is a palindrome.");
            } else {
                System.out.println(word + " is not a palindrome.");
            }
        }
    }
    public static boolean isPalindrome(String word) {
        int left = 0, right = word.length() - 1;
        while (left < right) {
            if (word.charAt(left) != word.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

OUTPUT:

```
run:
laiba is not a palindrome.
sara is not a palindrome.
racecar is a palindrome.
MOM is a palindrome.
BUILD SUCCESSFUL (total time: 4 seconds)
```

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

CODE:

```
package lab4.pkg015;
import java.util.Arrays;
public class Lab4015 {
    public static void main(String[] args) {
        // TASK 4
        int[] numbers = {04, 15, 19, 277, 44, 392};
        int evenCount = 0;
        int oddCount = 0;

        for (int num : numbers) {
            if (num % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
        }
        System.out.println("array : "+ Arrays.toString(numbers));
        System.out.println("Even numbers: " + evenCount);
        System.out.println("Odd numbers: " + oddCount);
    }
}
```

OUTPUT:

```
run:
array : [4, 15, 19, 277, 44, 392]
Even numbers: 3
Odd numbers: 3
BUILD SUCCESSFUL (total time: 1 second)
```

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

CODE:

```
package lab4.pkg015;
import java.util.ArrayList;
public class Lab4015 {
    public static void main(String[] args) {
        int[] array1 = {15, 2, 39, 41, 5};
        int[] array2 = {39, 41, 5, 6, 27};
        int[] mergedArray = mergeAndRemoveDuplicates(array1, array2);
        System.out.println("array: ");
        for (int num : mergedArray) {
            System.out.print(num + " ");
        }
        System.out.println();
    }
    public static int[] mergeAndRemoveDuplicates(int[] array1, int[] array2) {
        ArrayList<Integer> resultList = new ArrayList<>();
        for (int num : array1) {
            if (!resultList.contains(num)) {
                resultList.add(num); }
        }
        for (int num : array2) {
            if (!resultList.contains(num)) {
                resultList.add(num); }
        }
        int[] resultArray = new int[resultList.size()];
        for (int i = 0; i < resultList.size(); i++) {
            resultArray[i] = resultList.get(i); }
        return resultArray;
}
    }
```

OUTPUT:

```
run:
array:
15 2 39 41 5 6 27
BUILD SUCCESSFUL (total time: 1 second)
```

HOME TASKS

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:

```
package lab4.pkg015;
public class LAB4 {
    public double[] arr;
    public LAB4(double[] arr) {
        if (arr.length != 7) {
            throw new IllegalArgumentException("Array size must be 7.");
        }
        this.arr = arr;
    }
    public double calSum() {
        double sum = 0;
        for (double num : arr) {
            sum += num;
        }
        return sum;
    }
    public double calMean() {
        return calSum() / arr.length;
    }
    public static void main(String[] args) {
        double[] array = {4, 2.3, 15, 4.1, 19, 6.2, 44};
        LAB4 lab4 = new LAB4(array);
        System.out.println("Sum: " + lab4.calSum());
        System.out.println("Mean: " + lab4.calMean());
    }
}
```

   OUTPUT:

```
run:
Sum: 94.6
Mean: 13.514285714285714
BUILD SUCCESSFUL (total time: 1 second)
```

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:

```java
package arraysplitter;
public class ArraySplitter {
    public static void splitArrayAtKey(int[] array, int key) {
        int index = -1;
        for (int i = 0; i < array.length; i++) {
            if (array[i] == key) {
                index = i;
                break; // Stop once the key is found
            }
        }
        if (index != -1) {
            int[] firstPart = new int[index];
            int[] secondPart = new int[array.length - index - 1];
            for (int i = 0; i < index; i++) {
                firstPart[i] = array[i];
            }
            for (int i = index + 1; i < array.length; i++) {
                secondPart[i - index - 1] = array[i];
            }
            System.out.println("First part of the array:");
            printArray(firstPart);
            System.out.println("Second part of the array:");
            printArray(secondPart);
        } else {
            System.out.println("Key not found in the array.");
        }
    }
    public static void printArray(int[] array) {
        for (int i : array) {
            System.out.print(i + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] myArray = {1, 3, 5,4,6,8, 7,4,5,6, 9, 11};
        int key = 8;
        splitArrayAtKey(myArray, key);  // Example usage
    }
```

OUTPUT:

```
run:
First part of the array:
1 3 5 4 6
Second part of the array:
7 4 5 6 9 11
BUILD SUCCESSFUL (total time: 1 second)
```

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:

```java
import java.util.Arrays;
public class Task3 {
    public static boolean hasCombinationSum(int[] array, int target) {
        Arrays.sort(array);
        int left = 0, right = array.length - 1;
        while (left < right) {
            int sum = array[left] + array[right];
            if (sum == target) {
                System.out.println("Combination found: [" + array[left] + ", " + array[right] + "]");
                return true;
            } else if (sum < target) {
                left++;
            } else {
                right--;
            }
        }
        System.out.println("No combination found that sums to " + target);
        return false;
    }
    public static void main(String[] args) {
        int[] array = {10, 1, 2, 7, 6, 5};
        int target = 8;
        hasCombinationSum(array, target);
    }
}
```

OUTPUT:

```
run:
Combination found: [1, 7]
BUILD SUCCESSFUL (total time: 2 seconds)
```

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:

```java
public class MissingNumber {
    public static int findMissingNumber(int[] array, int n) {
        int expectedSum = n * (n + 1) / 2;
        int actualSum = 0;
        for (int i = 0; i < array.length; i++) {
            actualSum += array[i];
        }
        return expectedSum - actualSum;
    }
    public static void main(String[] args) {
        int[] array = {0, 2, 3, 4, 5};
        int n = 5;
        System.out.println("The missing number is: " + findMissingNumber(array, n));
    }
}
```

OUTPUT:

```
run:
The missing number is: 1
BUILD SUCCESSFUL (total time: 1 second)
```

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:

```
public class ZigzagSort {
    public static void zigzagSort(int[] array) {
        for (int i = 0; i < array.length - 1; i++) {
            if (i % 2 == 0) {
                if (array[i] > array[i + 1]) {
                    int temp = array[i];
                    array[i] = array[i + 1];
                    array[i + 1] = temp;
                }
            }
            else {
                if (array[i] < array[i + 1]) {
                    int temp = array[i];
                    array[i] = array[i + 1];
                    array[i + 1] = temp;
                }
            }
        }
    }
    public static void printArray(int[] array) {
        for (int i : array) {
            System.out.print(i + " "); }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] array = {4, 3, 7, 8, 6, 2, 1};
        System.out.println("Original array:");
        printArray(array);
        zigzagSort(array);
        System.out.println("Zigzag sorted array:");
        printArray(array);
    }
}
```

OUTPUT:

```
run:
Original array:
4 3 7 8 6 2 1
Zigzag sorted array:
3 7 4 8 2 6 1
BUILD SUCCESSFUL (total time: 1 second)
```