

LAB # 04

Arrays in JAVA

OBJECTIVE:

To understand arrays and their memory allocation.

LAB TASKS:

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

CODE:

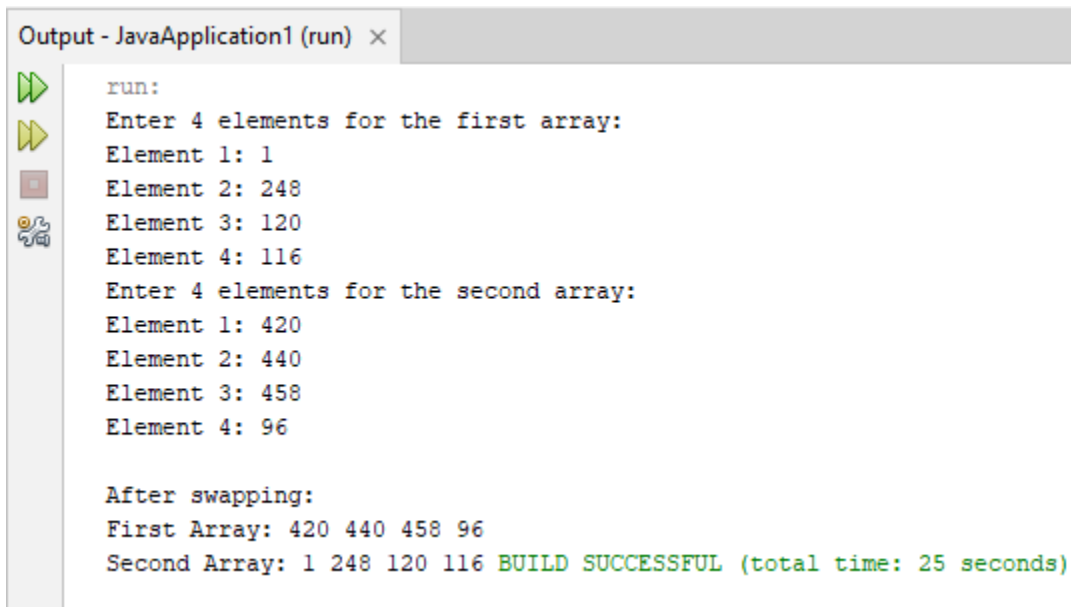
```
public static void main(String[] args) {  
    int[] array1 = new int[4];  
    int[] array2 = new int[4];  
    Scanner scanner = new Scanner(System.in);  
    System.out.println("Enter 4 elements for the first array:");  
    for (int i = 0; i < 4; i++) {  
        System.out.print("Element " + (i + 1) + ": ");  
        array1[i] = scanner.nextInt();  
        System.out.println("Enter 4 elements for the second array:");  
        for (int i = 0; i < 4; i++) {  
            System.out.print("Element " + (i + 1) + ": ");  
            array2[i] = scanner.nextInt();  
            for (int i = 0; i < 4; i++) {  
                int temp = array1[i];  
                array1[i] = array2[i];  
                array2[i] = temp;  
                System.out.println("\nAfter swapping:");  
                System.out.print("First Array: ");
```

Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

Data Structure & Algorithm Lab:04

```
for (int i = 0; i < 4; i++) {  
System.out.print(array1[i] + " ");  
  
System.out.print("\nSecond Array: ");  
  
for (int i = 0; i < 4; i++) {  
System.out.print(array2[i] + " ");  
  
scanner.close();
```

OUTPUT



The screenshot shows the output of a Java application. The window title is "Output - JavaApplication1 (run) x". The output text is as follows:

```
run:  
Enter 4 elements for the first array:  
Element 1: 1  
Element 2: 248  
Element 3: 120  
Element 4: 116  
Enter 4 elements for the second array:  
Element 1: 420  
Element 2: 440  
Element 3: 458  
Element 4: 96  
  
After swapping:  
First Array: 420 440 458 96  
Second Array: 1 248 120 116 BUILD SUCCESSFUL (total time: 25 seconds)
```

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

CODE:

```
public static void main(String[] args) {  
    int[] array1 = new int[4];  
    int[] array2 = new int[4];  
    Scanner scanner = new Scanner(System.in);  
    System.out.println("Enter 4 elements for the first array:");  
    for (int i = 0; i < 4; i++) {  
        System.out.print("Element " + (i + 1) + ": ");  
        array1[i] = scanner.nextInt();  
        System.out.println("Enter 4 elements for the second array:");  
        for (int i = 0; i < 4; i++) {  
            System.out.print("Element " + (i + 1) + ": ");  
            array2[i] = scanner.nextInt();  
            for (int i = 0; i < 4; i++) {  
                int temp = array1[i];  
                array1[i] = array2[i];  
                array2[i] = temp;  
                System.out.println("\nAfter swapping:");  
                System.out.print("First Array: ");  
                for (int i = 0; i < 4; i++) {  
                    System.out.print(array1[i] + " ");  
                    System.out.print("\nSecond Array: ");  
                    for (int i = 0; i < 4; i++) {  
                        System.out.print(array2[i] + " ");  
                    }  
                }  
            }  
        }  
    }  
}
```

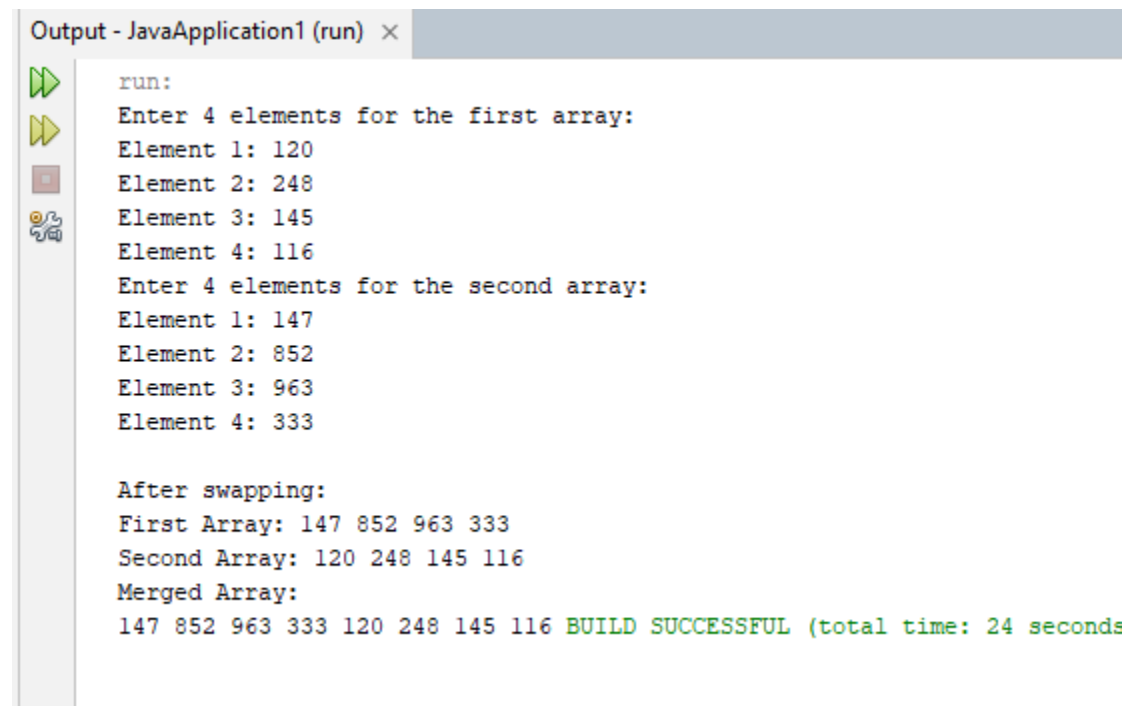
```
int[] mergedArray = mergeArrays(array1, array2);

System.out.println("\nMerged Array:");

for (int value : mergedArray) {
System.out.print(value + " ");

    public static int[] mergeArrays(int[] array1, int[] array2) {
int[] merged = new int[array1.length + array2.length];
for (int i = 0; i < array1.length; i++) {
merged[i] = array1[i];
for (int i = 0; i < array2.length; i++) {
merged[array1.length + i] = array2[i];
return merged
```

OUTPUT:



The screenshot shows the output of a Java application. The window title is "Output - JavaApplication1 (run)". The output text is as follows:

```
run:
Enter 4 elements for the first array:
Element 1: 120
Element 2: 248
Element 3: 145
Element 4: 116
Enter 4 elements for the second array:
Element 1: 147
Element 2: 852
Element 3: 963
Element 4: 333

After swapping:
First Array: 147 852 963 333
Second Array: 120 248 145 116
Merged Array:
147 852 963 333 120 248 145 116 BUILD SUCCESSFUL (total time: 24 seconds)
```

Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

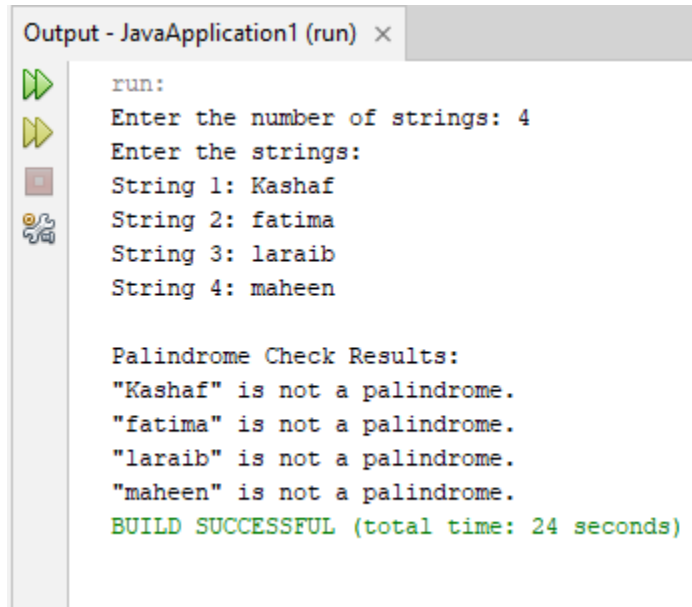
Data Structure & Algorithm Lab:04

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

CODE:

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.print("Enter the number of strings: ");  
    int n = scanner.nextInt();  
    scanner.nextLine(); // Consume the leftover newline  
    String[] strings = new String[n];  
    System.out.println("Enter the strings:");  
    for (int i = 0; i < n; i++) {  
        System.out.print("String " + (i + 1) + ": ");  
        strings[i] = scanner.nextLine();  
        System.out.println("\nPalindrome Check Results:");  
        for (String str : strings) {  
            if (isPalindrome(str)) {  
                System.out.println "\"" + str + "\" is a palindrome.");  
            } else {  
                System.out.println "\"" + str + "\" is not a palindrome.");  
            }  
        }  
    }  
    public static boolean isPalindrome(String str) {  
        int left = 0;  
        int right = str.length() - 1;  
        while (left < right) {  
            if (str.charAt(left) != str.charAt(right)) {  
                return false;  
            }  
            left++;  
            right--;  
        }  
        return true;  
    }  
}
```

OUTPUT:



```
Output - JavaApplication1 (run) x
run:
Enter the number of strings: 4
Enter the strings:
String 1: Kashaf
String 2: fatima
String 3: laraib
String 4: maheen

Palindrome Check Results:
"Kashaf" is not a palindrome.
"fatima" is not a palindrome.
"laraib" is not a palindrome.
"maheen" is not a palindrome.
BUILD SUCCESSFUL (total time: 24 seconds)
```

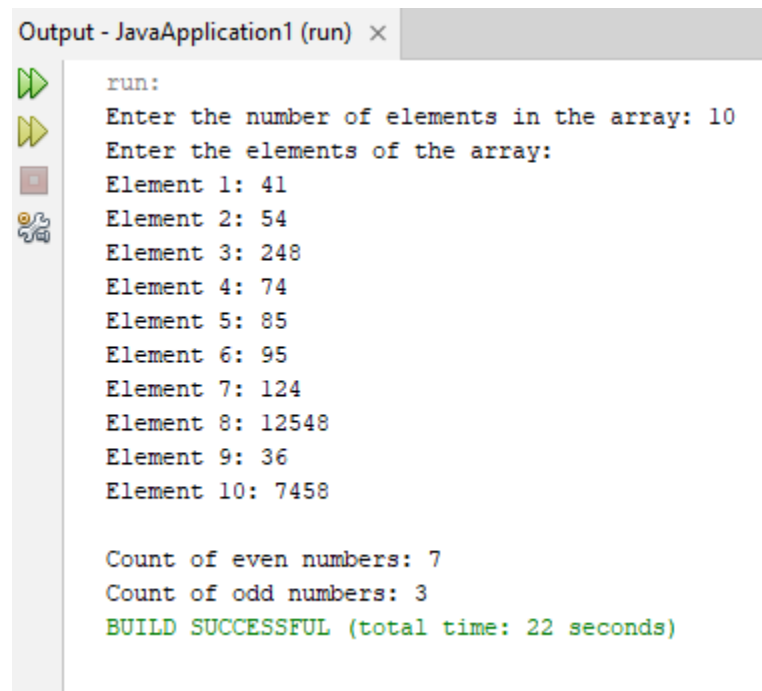
Given an array of integers, count how many numbers are even and how many are odd.

CODE:

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter the number of elements in the array: ");
    int n = scanner.nextInt();
    int[] numbers = new int[n];
    System.out.println("Enter the elements of the array:");
    for (int i = 0; i < n; i++) {
        System.out.print("Element " + (i + 1) + ": ");
        numbers[i] = scanner.nextInt();
        int evenCount = 0, oddCount = 0;
        for (int num : numbers) {
            if (num % 2 == 0) {
```

```
        evenCount++;  
  
    else {  
  
        oddCount++;  
  
        System.out.println("\nCount of even numbers: " + evenCount);  
  
        System.out.println("Count of odd numbers: " + oddCount);  
    }  
}
```

OUTPUT:



The screenshot shows the output of a Java application. It starts with a 'run:' prompt, followed by user input for the number of elements (10) and the elements themselves (41, 54, 248, 74, 85, 95, 124, 12548, 36, 7458). The program then outputs the count of even numbers (7) and odd numbers (3), and ends with a 'BUILD SUCCESSFUL' message.

```
run:  
Enter the number of elements in the array: 10  
Enter the elements of the array:  
Element 1: 41  
Element 2: 54  
Element 3: 248  
Element 4: 74  
Element 5: 85  
Element 6: 95  
Element 7: 124  
Element 8: 12548  
Element 9: 36  
Element 10: 7458  
  
Count of even numbers: 7  
Count of odd numbers: 3  
BUILD SUCCESSFUL (total time: 22 seconds)
```

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

CODE:

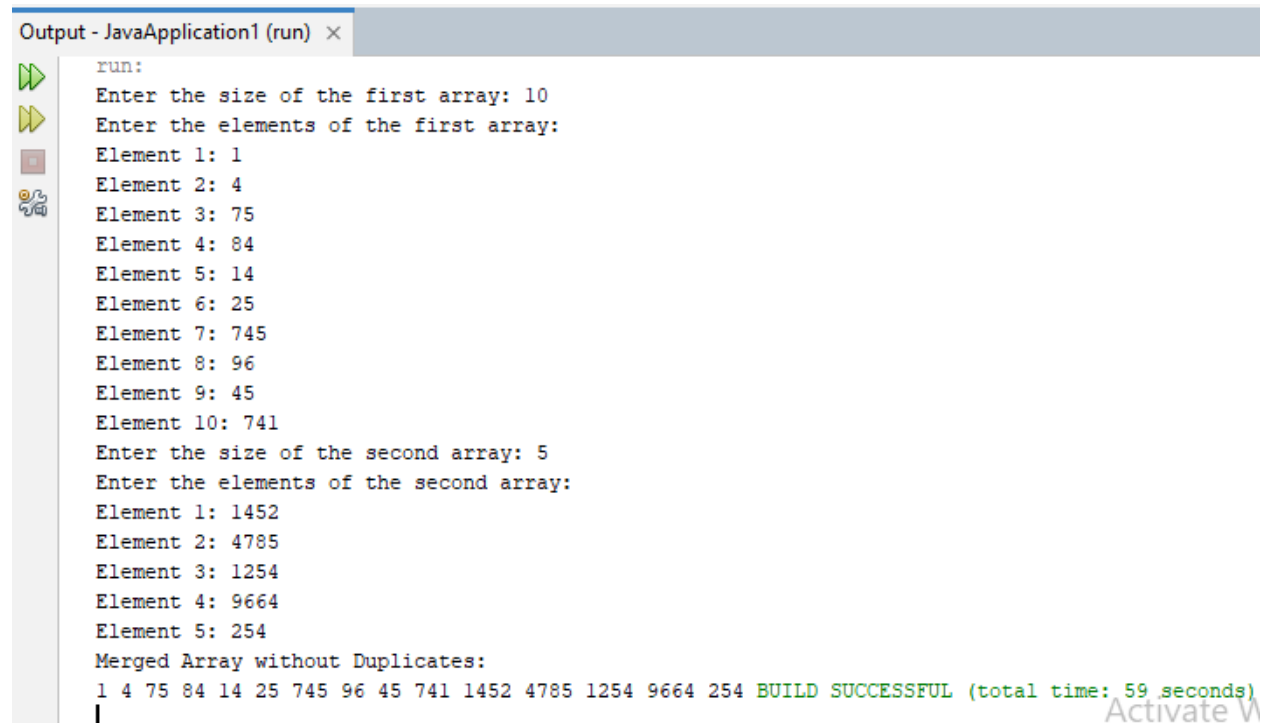
```
public static void main(String[] args) {  
  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Enter the size of the first array: ");  
  
    int n1 = scanner.nextInt();  
  
    int[] array1 = new int[n1];  
  
    System.out.println("Enter the elements of the first array:");  
}
```

```
    for (int i = 0; i < n1; i++) {  
System.out.print("Element " + (i + 1) + ": ");  
array1[i] = scanner.nextInt();  
    System.out.print("Enter the size of the second array: ");  
    int n2 = scanner.nextInt();  
    int[] array2 = new int[n2];  
    System.out.println("Enter the elements of the second array:");  
    for (int i = 0; i < n2; i++) {  
System.out.print("Element " + (i + 1) + ": ");  
array2[i] = scanner.nextInt();  
    int[] mergedArray = mergeArrays(array1, array2);  
    int[] uniqueArray = removeDuplicates(mergedArray);  
    System.out.println("Merged Array without Duplicates:");  
    for (int num : uniqueArray) {  
System.out.print(num + " ");  
    scanner.close();  
    public static int[] mergeArrays(int[] array1, int[] array2) {  
    int[] merged = new int[array1.length + array2.length];  
    for (int i = 0; i < array1.length; i++) {  
merged[i] = array1[i];  
    for (int i = 0; i < array2.length; i++) {  
merged[array1.length + i] = array2[i];  
    return merged;  
    public static int[] removeDuplicates(int[] array) {  
    int[] tempArray = new int[array.length];  
    int uniqueCount = 0;
```



```
for (int i = 0; i < array.length; i++) {  
    boolean isDuplicate = false;  
    for (int j = 0; j < uniqueCount; j++) {  
        if (array[i] == tempArray[j]) {  
            isDuplicate = true;  
            break;  
        }  
    }  
    if (!isDuplicate) {  
        tempArray[uniqueCount] = array[i];  
        uniqueCount++;  
    }  
    int[] result = new int[uniqueCount];  
    for (int i = 0; i < uniqueCount; i++) {  
        result[i] = tempArray[i];  
    }  
}
```

OUTPUT:



The screenshot shows a Java application window titled "Output - JavaApplication1 (run) x". The output text is as follows:

```
run:  
Enter the size of the first array: 10  
Enter the elements of the first array:  
Element 1: 1  
Element 2: 4  
Element 3: 75  
Element 4: 84  
Element 5: 14  
Element 6: 25  
Element 7: 745  
Element 8: 96  
Element 9: 45  
Element 10: 741  
Enter the size of the second array: 5  
Enter the elements of the second array:  
Element 1: 1452  
Element 2: 4785  
Element 3: 1254  
Element 4: 9664  
Element 5: 254  
Merged Array without Duplicates:  
1 4 75 84 14 25 745 96 45 741 1452 4785 1254 9664 254 BUILD SUCCESSFUL (total time: 59 seconds)
```

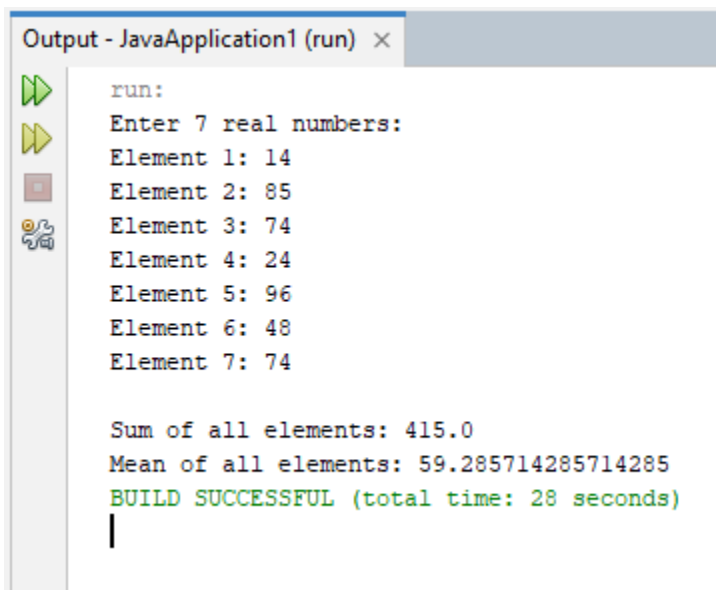
HOME TASKS:

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:

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    double[] numbers = new double[7]; // Array to store 7 real numbers  
  
    double sum = 0;  
  
    System.out.println("Enter 7 real numbers:");  
  
    for (int i = 0; i < 7; i++) {  
        System.out.print("Element " + (i + 1) + ": ");  
        numbers[i] = scanner.nextDouble();  
        sum += numbers[i]; // Calculate sum while reading input  
    }  
  
    double mean = sum / numbers.length; // Calculate mean  
  
    System.out.println("\nSum of all elements: " + sum);  
  
    System.out.println("Mean of all elements: " + mean);  
}
```

OUTPUT:



```
Output - JavaApplication1 (run) x  
run:  
Enter 7 real numbers:  
Element 1: 14  
Element 2: 85  
Element 3: 74  
Element 4: 24  
Element 5: 96  
Element 6: 48  
Element 7: 74  
  
Sum of all elements: 415.0  
Mean of all elements: 59.285714285714285  
BUILD SUCCESSFUL (total time: 28 seconds)  
|
```

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:

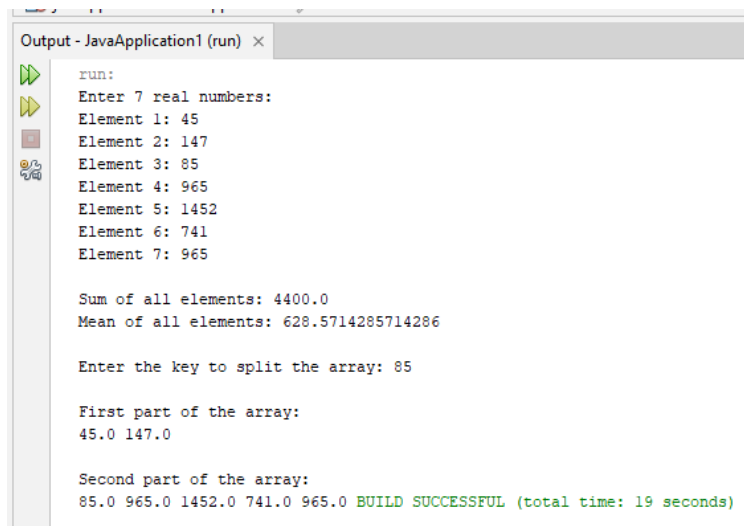
```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    double[] numbers = new double[7];  
  
    double sum = 0;  
  
    System.out.println("Enter 7 real numbers:");  
  
    for (int i = 0; i < 7; i++) {  
        System.out.print("Element " + (i + 1) + ": ");  
        numbers[i] = scanner.nextDouble();  
        sum += numbers[i];  
  
        double mean = sum / numbers.length;  
  
        System.out.println("\nSum of all elements: " + sum);  
        System.out.println("Mean of all elements: " + mean);  
  
        System.out.print("\nEnter the key to split the array: ");  
  
        double key = scanner.nextDouble();  
  
        double[][] splitArrays = splitArray(numbers, key);  
  
        if (splitArrays != null) {  
            System.out.println("\nFirst part of the array:");  
  
            for (double num : splitArrays[0]) {  
                System.out.print(num + " ")  
            }  
  
            System.out.println("\n\nSecond part of the array:");  
  
            for (double num : splitArrays[1]) {  
                System.out.print(num + " ");  
            }  
  
            else {  
                System.out.println("\nKey not found in the array. No split performed.");  
            }  
        }  
    }  
}
```

Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

Data Structure & Algorithm Lab:04

```
public static double[][] splitArray(double[] array, double key) {  
    int keyIndex = -1;  
    for (int i = 0; i < array.length; i++) {  
        if (array[i] == key) {  
            keyIndex = i;  
            break;  
        }  
        if (keyIndex == -1) {  
            return null;  
        }  
        double[] firstPart = new double[keyIndex];  
        for (int i = 0; i < keyIndex; i++) {  
            firstPart[i] = array[i];  
        }  
        double[] secondPart = new double[array.length - keyIndex];  
        for (int i = keyIndex; i < array.length; i++) {  
            secondPart[i - keyIndex] = array[i];  
        }  
        return new double[][]{firstPart, secondPart};  
    }  
}
```

OUTPUT:



```
run:  
Enter 7 real numbers:  
Element 1: 45  
Element 2: 147  
Element 3: 85  
Element 4: 965  
Element 5: 1452  
Element 6: 741  
Element 7: 965  
  
Sum of all elements: 4400.0  
Mean of all elements: 628.5714285714286  
  
Enter the key to split the array: 85  
  
First part of the array:  
45.0 147.0  
  
Second part of the array:  
85.0 965.0 1452.0 741.0 965.0 BUILD SUCCESSFUL (total time: 19 seconds)
```

Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

Data Structure & Algorithm Lab:04

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:

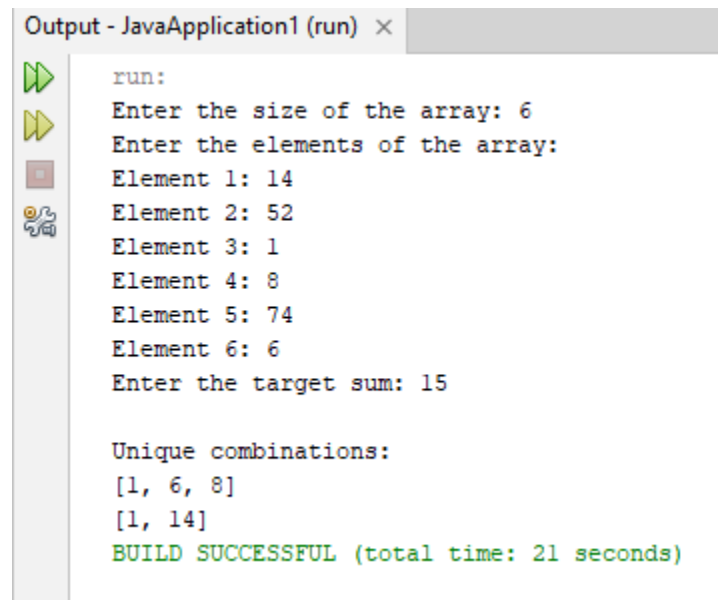
```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.print("Enter the size of the array: ");  
    int n = scanner.nextInt();  
    int[] array = new int[n];  
    System.out.println("Enter the elements of the array:");  
    for (int i = 0; i < n; i++) {  
        System.out.print("Element " + (i + 1) + ": ");  
        array[i] = scanner.nextInt();  
  
        System.out.print("Enter the target sum: ");  
        int target = scanner.nextInt();  
        List<List<Integer>> result = findCombinations(array, target);  
        System.out.println("\nUnique combinations:");  
        for (List<Integer> combination : result) {  
            System.out.println(combination);  
        }  
    }  
  
    public static List<List<Integer>> findCombinations(int[] array, int target) {  
        List<List<Integer>> result = new ArrayList<>();  
        Arrays.sort(array);  
        backtrack(result, new ArrayList<>(), array, target, 0);  
        return result;  
    }  
}
```

Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

Data Structure & Algorithm Lab:04

```
private static void backtrack(List<List<Integer>> result, List<Integer> tempList, int[] array, int
target, int start) {
    if (target == 0) {
        result.add(new ArrayList<>(tempList));
        return;
    }
    for (int i = start; i < array.length; i++) {
        if (array[i] > target) {
            break;
        }
        tempList.add(array[i]);
        backtrack(result, tempList, array, target - array[i], i + 1);
        tempList.remove(tempList.size() - 1);
    }
}
```

OUTPUT:



```
Output - JavaApplication1 (run) ×
run:
Enter the size of the array: 6
Enter the elements of the array:
Element 1: 14
Element 2: 52
Element 3: 1
Element 4: 8
Element 5: 74
Element 6: 6
Enter the target sum: 15

Unique combinations:
[1, 6, 8]
[1, 14]
BUILD SUCCESSFUL (total time: 21 seconds)
```

Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

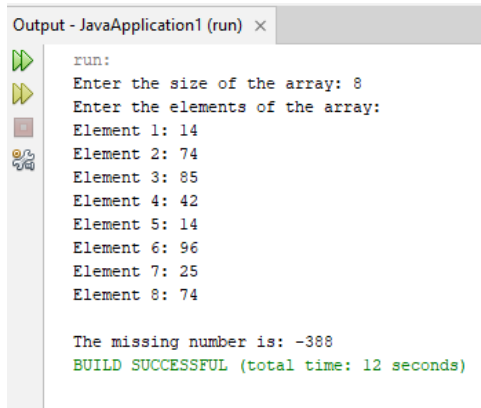
Data Structure & Algorithm Lab:04

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:

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Enter the size of the array: ");  
  
    int n = scanner.nextInt();  
  
    int[] array = new int[n];  
  
    System.out.println("Enter the elements of the array:");  
  
    for (int i = 0; i < n; i++) {  
        System.out.print("Element " + (i + 1) + ": ");  
        array[i] = scanner.nextInt();  
  
        int missingNumber = findMissingNumber(array, n);  
  
        System.out.println("\nThe missing number is: " + missingNumber);  
  
        public static int findMissingNumber(int[] array, int n) {  
  
            int expectedSum = n * (n + 1) / 2;  
  
            int actualSum = 0;  
  
            for (int num : array) {  
                actualSum += num;  
            }  
        }  
    }  
}
```

OUTPUT:



```
Output - JavaApplication1 (run) ×  
run:  
Enter the size of the array: 8  
Enter the elements of the array:  
Element 1: 14  
Element 2: 74  
Element 3: 85  
Element 4: 42  
Element 5: 14  
Element 6: 96  
Element 7: 25  
Element 8: 74  
  
The missing number is: -388  
BUILD SUCCESSFUL (total time: 12 seconds)
```

Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

Data Structure & Algorithm Lab:04

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 static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Enter the size of the array: ");  
    int n = scanner.nextInt();  
    int[] array = new int[n];  
    System.out.println("Enter the elements of the array:");  
    for (int i = 0; i < n; i++) {  
        System.out.print("Element " + (i + 1) + ": ");  
        array[i] = scanner.nextInt();  
        zigzagSort(array);  
        System.out.println("\nArray in zigzag pattern:");  
        for (int num : array) {  
            System.out.print(num + " ");  
        }  
        public static void zigzagSort(int[] array) {  
            boolean less = true;  
            for (int i = 0; i < array.length - 1; i++) {  
                if (less) {  
                    if (array[i] > array[i + 1]) {  
                        swap(array, i, i + 1)  
                    }  
                }  
                else {  
                    if (array[i] < array[i + 1]) {  
                        swap(array, i, i + 1);  
                    }  
                }  
            }  
        }  
    }  
}
```


Name: Kashaf Khan
Roll Number: 2023F-BSE-248
Section: A

Data Structure & Algorithm Lab:04

```
less = !less;
```

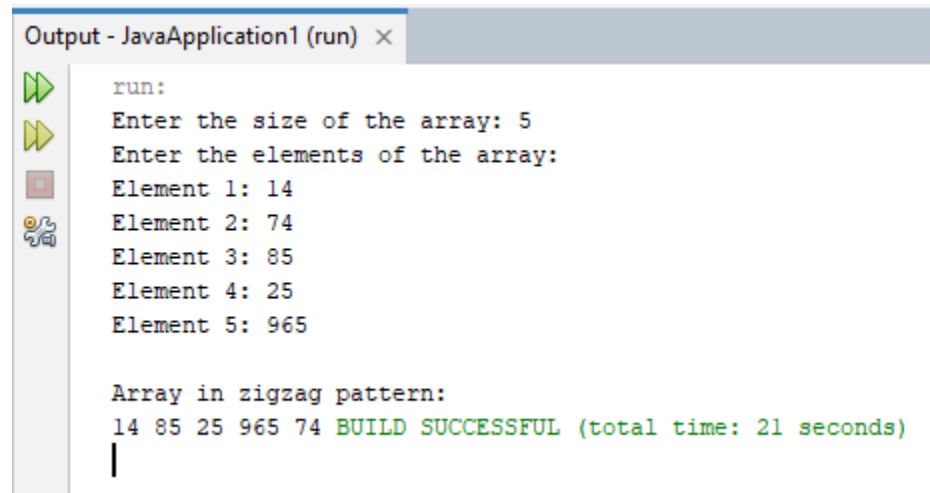
```
    public static void swap(int[] array, int i, int j) {
```

```
        int temp = array[i];
```

```
        array[i] = array[j];
```

```
        array[j] = temp;
```

OUTPUT:



```
Output - JavaApplication1 (run) ×
run:
Enter the size of the array: 5
Enter the elements of the array:
Element 1: 14
Element 2: 74
Element 3: 85
Element 4: 25
Element 5: 965

Array in zigzag pattern:
14 85 25 965 74 BUILD SUCCESSFUL (total time: 21 seconds)
|
```