NOTE: Please test all methods you have created by invoking it from main method

1. Write a Java Method to input n integer values from user, store it into an array and print all the elements.
2. Write a Java Method to sum values of an array input from user. Also calculate the average value of the elements.
3. Write a Java Method to test if an array contains a specific value. (Linear Search).
4. Write a Java Method to find the index of an array element.
5. Write a Java Method to remove a specific element from an array.
6. Write a Java Method to copy an array by iterating the array.
7. Write a Java Method to insert an element (specific position) into an array.
8. Write a Java Method to find the maximum and minimum value of an array.
9. Write a Java Method to get the difference between the largest and smallest values in an array of integers. The length of the array must be 1 and above.
10. Write a Java Method to reverse an array of integer values.
11. Write a Java Method to find the duplicate values of an array of integer values.
12. Write a Java Method to find the common elements between two arrays of integers.
13. Write a Java Method to find the second largest element in an array.
14. Write a java program to take a 2d array input from user and print the sum of each row, column and total sum in as shown below:

input:

1 3 5

2 4 6

output:

1 3 5 9

2 4 6 12

3 7 11 21

1. Write a Java Method to add two matrices of the same size.
2. Write a Java Method to input two matrices order, check if the order is eligible to multiply, if yes input the two matrices, multiply them and display the result.
3. Write a Java Method to input a matrix A and create its transpose matrix AT and print the result.
4. Write a Java Method that sorts an String array input by user in descending alphabetic order.
5. Write a Java Method to find all pairs of elements in an array whose sum is equal to a specified number.
6. Write a Java Method to test the equality of two arrays.
7. Write a Java Method to find the number of even and odd integers in a given array of integers.
8. Write a Java Method to compute the average value of an array of integers except the largest and smallest values.
9. Write a Java Method to check if an array of integers without 0 and -1.
10. Write a Java Method to check if an array of integers contains two specified elements 65 and 77.
11. Write a Java Method to remove the duplicate elements of a given array and return the new length of the array.  
    Sample array: [20, 20, 30, 40, 50, 50, 50]  
    After removing the duplicate elements the Method should return 4 as the new length of the array.
12. Write a Java Method to find the sum of the two elements of a given array which is equal to a given integer.  
    Sample array: [1,2,4,5,6]  
    Target value: 6.
13. Write a Java Method to print all the LEADERS in the array.     
    Note: An element is leader if it is greater than all the elements to its right side.
14. Write a Java Method to find smallest and second smallest elements of a given array.
15. Write a Java Method to segregate all 0s on left side and all 1s on right side of a given array of 0s and 1s.
16. Write a Java Method to cyclically rotate a given array clockwise by one.
17. Write a Java Method to sort an array of positive integers of a given array, in the sorted array the value of the first element should be maximum, second value should be minimum value, third should be second maximum, fourth second be second minimum and so on.
18. Write a Java Method to separate even and odd numbers of a given array of integers. Put all even numbers first, and then odd numbers.
19. Write a Java Method to sort a given binary array in linear times.     
    From Wikipedia,  
    Linear time: An algorithm is said to take linear time, or O(n) time, if its time complexity is O(n). Informally, this means that the running time increases at most linearly with the size of the input. More precisely, this means that there is a constant c such that the running time is at most cn for every input of size n. For example, a procedure that adds up all elements of a list requires time proportional to the length of the list, if the adding time is constant, or, at least, bounded by a constant.  
    Linear time is the best possible time complexity in situations where the algorithm has to sequentially read its entire input. Therefore, much research has been invested into discovering algorithms exhibiting linear time or, at least, nearly linear time. This research includes both software and hardware methods. There are several hardware technologies which exploit parallelism to provide this. An example is content-addressable memory. This concept of linear time is used in string matching algorithms such as the Boyer–Moore algorithm and Ukkonen's algorithm.

Example:  
Input :  
b\_nums[] = { 0, 1, 1, 0, 1, 1, 0, 1, 0, 0 }  
Output:  
After sorting: [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]

1. Given two sorted arrays A and B of size p and q, write a Java Method to merge elements of A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B with remaining elements.

Example:  
Input :  
int[] A = { 1, 5, 6, 7, 8, 10 }  
int[] B = { 2, 4, 9 }  
Output:  
Sorted Arrays:  
A: [1, 2, 4, 5, 6, 7]  
B: [8, 9, 10]

1. Write a Java Method to shuffle a given array of integers.

Example:  
Input :  
nums = { 1, 2, 3, 4, 5, 6 }  
Output:  
Shuffle Array: [4, 2, 6, 5, 1, 3]

1. Write a Java Method to replace each element of the array with product of every other element in a given array of integers.

Example:  
Input :  
nums1 = { 1, 2, 3, 4, 5, 6, 7}  
nums2 = {0, 1, 2, 3, 4, 5, 6, 7}  
Output:  
Array with product of every other element:  
[5040, 2520, 1680, 1260, 1008, 840, 720]  
Array with product of every other element:  
[5040, 0, 0, 0, 0, 0, 0, 0]

1. Write a Java Method to find maximum difference between two elements in a given array of integers such that smaller element appears before larger element.

Example:  
Input :  
nums = { 2, 3, 1, 7, 9, 5, 11, 3, 5 }  
Output:  
The maximum difference between two elements of the said array elements  
10