Simple Array Programming Practice Tasks (Java)

1. Write a program to sort a numeric array and a string array.
2. Write a program to sum values of an array.
3. Write a program to print the following grid.
Expected Output :
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -

4. Write a program to calculate the average value of array elements.

5. Write a program to test if an array contains a specific value.

6. Write a program to find the index of an array element.

7. Write a program to remove a specific element from an array.

8. Write a program to copy an array by iterating the array.

9. Write a program to insert an element (specific position) into an array.

10. Write a program to find the maximum and minimum value of an array.

11. Write a program to reverse an array of integer values.

12. Write a program to find the duplicate values of an array of integer values.

13. Write a program to find the duplicate values of an array of string values.

14. Write a program to find the common elements between two arrays (string values).

15. Write a program to find the common elements between two arrays of integers.

16. Write a program to remove duplicate elements from an array.

17. Write a program to find the second largest element in an array.

18. Write a program to find the second smallest element in an array.

19. Write a program to add two matrices of the same size.

20. Write a program to convert an array to ArrayList.

21. Write a program to convert an ArrayList to an array.

22. Write a program to find all pairs of elements in an array whose sum is equal to a specified number.

23. Write a program to test the equality of two arrays.

24. Write a program to find a missing number in an array.

25. Write a program to find common elements from three sorted (in non-decreasing order) arrays.

26. Write a program to move all 0's to the end of an array. Maintain the relative order of the other (non-zero) array elements.

27. Write a program to find the number of even and odd integers in a given array of integers.

28. Write a program 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.

29. Write a program to compute the average value of an array of integers except the largest and smallest values.

30. Write a program to check if an array of integers without 0 and -1.

31. Write a program to check if the sum of all the 10's in the array is exactly 30. Return false if the condition does not satisfy, otherwise true.

32. Write a program to check if an array of integers contains two specified elements 65 and 77.

33. Write a program 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 program should return 4 as the new length of the array.

34. Write a program to find the length of the longest consecutive elements sequence from a given unsorted array of integers.
Sample array: [49, 1, 3, 200, 2, 4, 70, 5]
The longest consecutive elements sequence is [1, 2, 3, 4, 5], therefore the program will return its length 5.

35. Write a program 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.

36. Write a program to find all the unique triplets such that sum of all the three elements [x, y, z (x ≤ y ≤ z)] equal to a specified number.
Sample array: [1, -2, 0, 5, -1, -4]
Target value: 2.

37. Write a program to create an array of its anti-diagonals from a given square matrix. Go to the editor

Example:
Input :
1 2
3 4
Output:
[
[1],
[2, 3],
[4]
]

38. Write a program to get the majority element from a given array of integers containing duplicates.
Majority element: A majority element is an element that appears more than n/2 times where n is the size of the array.

39. Write a program 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.

40. Write a program to find the two elements from a given array of positive and negative numbers such that their sum is closest to zero.

41. Write a program to find smallest and second smallest elements of a given array.

42. Write a program to segregate all 0s on left side and all 1s on right side of a given array of 0s and 1s.

43. Write a program to find all combination of four elements of a given array whose sum is equal to a given value.

44. Write a program to count the number of possible triangles from a given unsorted array of positive integers.

Note: The triangle inequality states that the sum of the lengths of any two sides of a triangle must be greater than or equal to the length of the third side.

45. Write a program to cyclically rotate a given array clockwise by one.

46. Write a program to check whether there is a pair with a specified sum of a given sorted and rotated array.

47. Write a program to find the rotation count in a given rotated sorted array of integers.

48. Write a program to arrange the elements of a given array of integers where all negative integers appear before all the positive integers.

49. Write a program to arrange the elements of a given array of integers where all positive integers appear before all the negative integers.

50. Write a program 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.

51. Write a program to separate 0s on left side and 1s on right side of an array of 0s and 1s in random order.

52. Write a program to separate even and odd numbers of a given array of integers. Put all even numbers first, and then odd numbers.

53. Write a program to replace every element with the next greatest element (from right side) in a given array of integers. There is no element next to the last element, therefore replace it with -1.

54. Write a program to check if a given array contains a subarray with 0 sum.

Example:
Input :
nums1= { 1, 2, -2, 3, 4, 5, 6 }
nums2 = { 1, 2, 3, 4, 5, 6 }
nums3 = { 1, 2, -3, 4, 5, 6 }
Output:
Does the said array contain a subarray with 0 sum: true
Does the said array contain a subarray with 0 sum: false
Does the said array contain a subarray with 0 sum: true

55. Write a program to print all sub-arrays with 0 sum present in a given array of integers.

Example:
Input :
nums1 = { 1, 3, -7, 3, 2, 3, 1, -3, -2, -2 }
nums2 = { 1, 2, -3, 4, 5, 6 }
nums3= { 1, 2, -2, 3, 4, 5, 6 }
Output:
Sub-arrays with 0 sum : [1, 3, -7, 3]
Sub-arrays with 0 sum : [3, -7, 3, 2, 3, 1, -3, -2]

Sub-arrays with 0 sum : [1, 2, -3]
Sub-arrays with 0 sum : [2, -2]

56. Write a   program to sort a given binary array in linear times.

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]

57. Write a program to check if a sub-array is formed by consecutive integers from a given array of integers.

Example:
Input :
nums = { 2, 5, 0, 2, 1, 4, 3, 6, 1, 0 }
Output:
The largest sub-array is [1, 7]
Elements of the sub-array: 5 0 2 1 4 3 6

58. Given two sorted arrays A and B of size p and q, write a   program 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]

59. Write a program to find maximum product of two integers in a given array of integers.   Go to the editor

Example:
Input :
nums = { 2, 3, 5, 7, -7, 5, 8, -5 }
Output:
Pair is (7, 8), Maximum Product: 56

60. Write a program 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]

61. Write a program to rearrange a given array of unique elements such that every second element of the array is greater than its left and right elements.

Example:
Input :
nums= { 1, 2, 4, 9, 5, 3, 8, 7, 10, 12, 14 }
Output:
Array with every second element is greater than its left and right elements:
[1, 4, 2, 9, 3, 8, 5, 10, 7, 14, 12]

62. Write a program to find the equilibrium indices from a given array of integers.

Example:
Input :
nums = {-7, 1, 5, 2, -4, 3, 0}
Output:
Equilibrium indices found at : 3
Equilibrium indices found at : 6

63. Write a program 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]

64. Write a program to find Longest Bitonic Subarray in a given array.

A bitonic subarray is a subarray of a given array where elements are first sorted in increasing order, then in decreasing order. A strictly increasing or strictly decreasing subarray is also accepted as bitonic subarray.

Example:
Input :
nums = { 4, 5, 9, 5, 6, 10, 11, 9, 6, 4, 5 }
Output:
The longest bitonic subarray is [3,9]
Elements of the said sub-array: 5 6 10 11 9 6 4
The length of longest bitonic subarray is 7

65. Write a program 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

66. Write a program to find contiguous subarray within a given array of integers which has the largest sum.

In computer science, the maximum sum subarray problem is the task of finding a contiguous subarray with the largest sum, within a given one-dimensional array A[1...n] of numbers. Formally, the task is to find indices and with, such that the sum is as large as possible.

Example:
Input :
int[] A = {1, 2, -3, -4, 0, 6, 7, 8, 9}
Output:
The largest sum of contiguous sub-array: 30

67. Write a program to find subarray which has the largest sum in a given circular array of integers.

Example:
Input :
nums1 = { 2, 1, -5, 4, -3, 1, -3, 4, -1 }
nums2 = { 1, -2, 3, 0, 7, 8, 1, 2, -3 }
Output:
The sum of subarray with the largest sum is 6
The sum of subarray with the largest sum is 21

68. Write a program to create all possible permutations of a given array of distinct integers.

Example:
Input :
nums1 = {1, 2, 3, 4}
nums2 = {1, 2, 3}
Output:
Possible permutations of the said array:

[1, 2, 3, 4]
[1, 2, 4, 3]
....
[4, 1, 3, 2]
[4, 1, 2, 3]
Possible permutations of the said array:
[1, 2, 3]
[1, 3, 2]
...
[3, 2, 1]
[3, 1, 2]

69. Write a program to find minimum subarray sum of specified size in a given array of integers.

Example:
Input :
nums = { 1, 2, 3, 4, 5, 6, 7, 8, 9,10}
Output:
Sub-array size: 4
Sub-array from 0 to 3 and sum is: 10

70. Write a program to find the smallest length of a contiguous subarray of which the sum is greater than or equal to specified value. Return 0 instead.
Example:
Input :
nums = {1, 2, 3, 4, 6}
Output:
Minimum length of a contiguous subarray of which the sum is 8, 2

71. Write a program to form the largest number from a given list of non negative integers.

Example:
Input :
nums = {1, 2, 3, 0, 4, 6}
Output:
Largest number using the said array numbers: 643210

72. Write a program to find and print one continuous subarray (from a given array of integers) that if you only sort the said subarray in ascending order then the entire array will be sorted in ascending order.

Example:

Input :
nums1 = {1, 2, 3, 0, 4, 6}
nums2 = { 1, 3, 2, 7, 5, 6, 4, 8}
Output:
Continuous subarray:
1 2 3 0
Continuous subarray:
3 2 7 5 6 4

73. Write a program to sort a given array of distinct integers where all its numbers are sorted except two numbers.

Example:
Input :
nums1 = { 3, 5, 6, 9, 8, 7 }
nums2 = { 5, 0, 1, 2, 3, 4, -2 }
Output:
After sorting new array becomes: [3, 5, 6, 7, 8, 9]
After sorting new array becomes: [-2, 0, 1, 2, 3, 4, 5]

74. Write a program to find all triplets equal to a given sum in a unsorted array of integers.

Example:
Input :
nums = { 1, 6, 3, 0, 8, 4, 1, 7 }
Output:
Triplets of sum 7
(0 1 6)
(0 3 4)