Data Structure and Algorithms

Problem Set 3: Arrays and Strings

Date of issue: **Due Date:**

Problem 1) Create an array of 10 elements, input 10 elements and then search an element into array.
Input: - Enter 10 elements: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Enter element to be searched: 40
Output: - Element 40 found at loc: 4

Problem 2) Create an array of 50 elements, store 10 elements and then insert an element at the first location of array...
Input: - Enter 10 Elements: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Enter element: 1
Output: - Array after inserting 1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Problem 3) Create an array of 50 elements, store 10 element and then insert an element at the last location of array.
Input: - Enter 10 Elements: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Enter element: 11
Output: - Array after inserting 1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Problem 4) Create an array of 50 elements, store 10 elements and then insert an element at the desired location of array...
Input: - Enter 10 Elements: 1, 2, 3, 5, 6, 7, 8, 9, 10, 11
Enter element: 4
Enter location: 4
Output: - Array after inserting 4 at loc 4: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Problem 5) Write a C program that, given an array A[] of n numbers and another number x, determines whether or not there exist two elements in S whose sum is exactly x

Case 1) Input: Array 1, 4, 45, 6, 10, -8 X= 16
Output: Yes
Case 2) Input: Array 1, 4, 45, 6, 10, -8 X= 116
Output: No

Problem 6) Given an array of numbers, arrange in it in the form of single number such that the concluded number is the maximum.

Input: - 78, 92, 43, 9, 10
Output: - 998743210

Input: - 99, 8, 76, 45, 7, 33, 42
Output: - 998776544332

Problem 7) Majority Element: A majority element in an array A[] of size n is an element that appears more than n/2 times (and hence there is at most one such element)(Use **Moore's Voting Algorithm**)

Case 1) I/P: 3 3 4 2 4 4 2 4 4
O/P: 4
Case 2) I/P: 3 3 4 2 4 4 2 4
O/P: NONE

Problem 8) Write an efficient C program to find the sum of contiguous sub-array within a one-dimensional array of numbers which has the largest sum. (Use **Kadane's Algorithm**)

Input: -2, -3, 4, -1, -2, 1, 5, -3

Output: Max Sum is =7 (4+ (-1) + (-2) +1+5)