National Institute of Technology Calicut
Department of Computer Science and Engineering
Third Semester B. Tech.(CSE)
CS2092D Programming Laboratory
Assignment #2

Submission deadline (on or before):

- 01.09.2021, 9:00 AM

Policies for Submission and Evaluation:

- You must submit your assignment in the Eduserver course page, on or before the submission deadline.
- Ensure that your programs will compile and execute without errors using gcc compiler.
- During the evaluation, failure to execute programs without compilation errors may lead to zero marks for that evaluation.
- Detection of ANY malpractice related to the lab course can lead to awarding an F grade in the course.

Naming Conventions for Submission

- Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar, .tar, .gz). The name of this file must be

  `ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>.zip`

  (Example: *ASSG1_BxxyyyyCS_LAXMAN.zip*). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

- The source codes must be named as

  `ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>_<PROGRAM-NUMBER>.c`

  (For example: *ASSG1_BxxyyyyCS_LAXMAN_1.c*). If you do not conform to the above naming conventions, your submission might not be recognized by our automated tools, and hence will lead to a score of 0 marks for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

- Violation of academic integrity will be severely penalized. Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course. The department policy on academic integrity can be found at: http://cse.nitc.ac.in/sites/default/files/Academic-Integrity_new.pdf.

General Instructions

- Programs should be written in C language and compiled using gcc compiler**Submit the solutions to the questions through the submission link in Eduserver**.
- Check your programs with sufficiently large values of inputs with in the range as specified in the question.
- Global and/or static variables should not be used in your program.

QUESTIONS

1. Write a program to find the maximum sum that can be obtained by picking some non-empty subset of elements from an integer array A. If there are many such non-empty subsets, choose the one with the maximum number of elements. Print the maximum sum and the number of elements in the chosen subset.

   Input format:

   - The first line of the input contains an integer $n \in [1, 10^2]$, the size of the array A.
   - The second line lists the n elements in A, as space-separated integers in the range $[-1000, 1000]$.

   Output Format:

   - Print two space-separated integers, the maximum sum that can be obtained by choosing some subset and the maximum number of elements among all such subsets that have the same maximum sum.

   Sample Input1:
   5
   1 2 -4 -3 3

   Sample Output1: 6 3

   Sample Input2:
   5
   2 -6 -4 -3 -8

   Sample Output2: 2 1

2. Write a program to find all numbers in an array A with digits in non-decreasing order.

   Input format:

   - The first line of the input contains an integers, $n \in [0, 10^4]$, the size of the array A.
   - The second line lists the n elements in A, as space-separated integers in the range $[-1000, 1000]$.

   Output Format:

   - The output contains all numbers with digits in non-decreasing order. Print the result in their order of occurrence in the array A.
   - If there is no numbers with digits in non-decreasing order in A, print -1.

   Sample Input1:
   6
   43 333 123 59 492 321

   Sample Output1:
   333 123 59

   Sample Input2:
   5
   43 13 10 89 49 32

Sample Output1:
13 89 49

3. Write a program to search a given sub-string (A sub string is a contiguous sequence of characters within a string) in a given character array A. If the sub-string is present, delete all occurrences of the sub-string from the array and print the resulting array.

   Input format:

   - The first line of the input is a string with uppercase and lowercase characters $\in [A - Z, a - z]$.
   - The second line of the input is a sub-string with uppercase and lowercase characters $\in [A - Z, a - z]$.

   Output format:

   - The output is a string after the removal of all the sub-string occurrences from the array A.

   Sample Input1: Honey bee
   ee

   Sample Output1: Honey b

   Sample Input2: Aaab
   aa

   Sample Output1: Ab

4. Write a program to sort the elements in an array A (array indexing starts from 0) based on their positions. Sort the even positioned elements (index 0, 2, 4, ...) in non-decreasing sorted order and the odd positioned elements (index 1, 3, 5, ..) in non-increasing sorted order using Insertion sort. The program must contain the following functions.

   - InsertionSortAs(B) - A function that takes as input an array B that contain even-positioned elements and sorts it in non-decreasing sorted order using insertion sort.
   - InsertionSortDs(C) - A function that takes as input an array C that contain odd-positioned elements and sorts it in non-increasing sorted order using insertion sort.
   - Print(X) - A function that takes as input an array X and prints its contents in order, with a single space separating the elements. This function should only be called from the main() function.

   Input format:

   - The first line of the input contains an integer $n \in [0, 10^3]$, the size of the array A.
   - The second line lists the n elements in A, as space-separated integers in the range $[-1000, 1000]$.

   Output Format:

   - The first line of the output contains the even-positioned elements of A in non-decreasing sorted order, with a single space separating each element.

- The second line of the output contains the odd-positioned elements of A in non-increasing sorted order, with a single space separating each element.

Sample Input1:
8
12 8 14 35 42 2 43 27

Sample Output1:
12 14 42 43
35 27 8 2

Sample Input2:
7
1 1 2 2 3 3 4

Sample Output2:
1 2 3 4

4 3 2 1
3 2 1

5. Write a program to calculate the difference between maximum sum and minimum sum of $n - m$ elements of the given integer array A (array indexing starts from 0) where n is the size of the array and m is a random integer.

Input format:

- The first line of the input contains two integers $n \in [0, 10^3]$, the size of the array A and $m \in [0, 10^3]$, $m < n$.
- The second line lists the n elements in A, as space-separated integers in the range $[-1000, 1000]$.

Output Format:

- The output contains the difference between maximum sum and minimum sum of $n - m$ elements of the given integer array A

Sample Input1:
8 3
10 23 4 15 22 2 45 27

Sample Output1:
79

Sample Input2:
7 3
1 2 3 4 5 6 7

Sample Output2:
12