

Algorithm Lab (Course Code: MC504)

Assignment - 2

Submission Deadline: Within class timing, (21/01/2022)

Total Marks: 50

Instructions:

- Proper indentation is mandatory.
 - Program files **must** be compiled using **linux gcc compiler**.
 - **VERY IMPORTANT:** You must add comments whenever necessary, to make the code understandable.
 - Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
 - Take inputs from users. Make necessary assumptions if required.
 - **ANSWER FILE:** Source code: (file name) e.g. A2_Q1.c, A2_PP.c
 - **Compress all the source code in a single zip/rar file(e.g. Rollno_Name.zip) and Upload on Teams.**
 - **Each source code file must contain your name and roll no as comments.**
-

Q1.

Implement the insertion sort algorithm in C. Take the input array to be of homogeneous numeric type.

Q2.

Read a positive integer value, and compute the following sequence: If the number is even, halve it; if it is odd, multiply by 3 and add 1. Repeat this process until the value is 1, printing out each value. Finally print out how many of these operations you performed. If the input value is less than 1, print a message containing the word "Error".

Typical output might be:

Initial value is 9

Next value is 28

Next value is 14

Next value is 7
Next value is 22
Next value is 11
Next value is 34
Next value is 17
Next value is 52
Next value is 26
Next value is 13
Next value is 40
Next value is 20
Next value is 10
Next value is 5
Next value is 16
Next value is 8
Next value is 4
Next value is 2
Final value 1, number of steps 19.

*Also compute the time complexity for the code written by you in Big O notation.
Write this as a comment in your source code file.*

Q3.

Write a C program to delete all the duplicates in an integer array. Note that after deleting a duplicate all the numbers will be shifted one position to the left.

*Also compute the time complexity for the code written by you in Big O notation.
Write this as a comment in your source code file.*

PRACTICE PROBLEM (PP)

Given an array of integers containing duplicate elements. The task is to find the sum of all odd occurring elements in the given array. That is the sum of all such elements whose frequency is odd in the array.

Example:

Input : arr[] = {1, 1, 2, 2, 3, 3, 3}

Output :9

The odd occurring element is 3, and it's number occurrence is 3. Therefore sum of all 3's in the array = 9.

Input : arr[] = {10, 20, 30, 40, 40}

Output : 60

Elements with odd frequency are 10, 20 and 30.

Sum = 60