Programming and Data Structures Laboratory | 2021-22 Autumn semester, Section 20

Lab Test 1 | January 11, 2022
Full marks: 3 x 20 = 60
Time: 09:15 – 11:45 (2 hours 30 minutes)

Instructions

* There are three (3) problems in this test, each of 20 marks. The program for each problem must be written in an individual C source file. You should submit the following three plaintext C source files: **<roll number>_T1_1.c, <roll number>_T1_2.c,** and **<roll number>_T1_3.c**

* Submission must be through the course Moodle, before 11:45 (according to the Moodle clock). If you miss this deadline, then you need to email your submission to TA Anurag Roy (Email: anurag_roy@iitkgp.ac.in) within 12:00; there will be a penalty of 30 marks for such late submissions. No submission will be allowed after 12:00; any submission reaching the TA's mailbox after 12:00 (according to the receipt timestamp of the email) will be rejected.

* You are NOT supposed to take the help of any person / TA / book / online material during the test. Any malpractice / plagiarism will be penalised severely, with the minimum being awarding zero for the entire test.

* It is your responsibility to make your programs understandable, through meaningful variable names, indentation, comments (if necessary). Programs that are not understandable will be penalized.

Problem 1

Write a program that takes an integer *n* between 1 and 9 as input, and prints out on the screen a pattern similar to the following. An example is shown for *n* = 5; note that your program should work for all inputs *n* between 1 and 9.

```
55555
 4444
  333
   22
    1
   22
  333
 4444
55555
```

Problem 2

Given a set of distinct positive integers (i.e., no duplicate values), you need to find the subsets of the given set such that each subset contains exactly three consecutive integers. For instance:

- The set {100, 56, 5, 6, 102, 58, 101, 57, 7} has 3 such subsets {5, 6, 7}, {100, 101, 102} and {56, 57, 58}.
- The set {100, 56, 6, 102, 58, 101, 57, 7, 103} has 3 such subsets {100, 101, 102}, {101, 102, 103} and {56, 57, 58}.
- The set {100, 56, 104, 6, 101, 57, 7, 103} does not have any such subset.

Write a C program that asks the user to enter a set of numbers and stores the numbers in an array, and then prints the subsets containing exactly three consecutive integers. You can assume that the input set will contain at most 100 integers, and that the user will enter distinct, positive integers (no need to check for duplicates).

Problem 3

Consider two arrays A and B which store *n* and *m* (user inputs) integers respectively. Both arrays can contain positive and negative integers. Write a program that first asks the user the values of *n* and *m* (i.e., the number of elements in the two arrays) and then takes *n* and *m* integers as input from the user, and stores in the two arrays. You can assume that maximum value of *n* and *m* will be 50.

Consider another integer array C that is initially empty. Your program should copy values from the arrays A and B to array C in the following manner:
(i) Start reading the elements in A and copy the elements to C (in the same order), as long as there is no change in the sign of the elements in A.
(ii) Once there is a sign-change, start reading elements from B, and copy those to C (in the same order) as long as there is no change in the sign of the elements in B; once there is a sign-change in the elements of B, again resume with the elements in A.
(iii) Continue the above process till one or both of the arrays A and B is exhausted. Copy the rest of the elements of the other array (if any) into C. Your program should print out the array C.

Example interaction of the program with user:

Enter number of elements in array A: 7
Enter elements in array A: 1 2 3 -4 -5 6 7
Enter number of elements in array B: 5
Enter elements in array B: 11 12 -13 -14 -15
Array C: 1 2 3 11 12 -4 -5 -13 -14 -15 6 7

Enter number of elements in array A: 7
Enter elements in array A: 10 -6 12 13 -67 32 6
Enter number of elements in array B: 5
Enter elements in array B: 14 7 -12 32 -8
Array C: 10 14 7 -6 -12 12 13 32 -67 -8 32 6