CS19003: Programming & Data Structure Lab, Section 15
Autumn 2020-21
Lab Test 1, January 14, 2021

PART - 1

Time: 9-10 am to 10-20 am **Total Marks: 10**

Instructions (Read carefully)

1. **Your C programs must first contain comments with your name, roll no., and Labtest no. (=1), as done in class.**
2. **Name your C file LT1_1_<your roll no>.c for (For example LT1_1_20ME30006.c)**
3. **Submit through the links (Intermediate and Final) for PART 1 in moodle. MAKE SURE TO VERIFY YOUR SUBMISSION after final submission.**
4. **All instructions given in the slides of last class (January 7) to be followed strictly**

---

1. Consider a set of points on 2-d plane, so each point has a x-coordinate and a y-coordinate. For any two points P1 = (x1, y1) and P2 = (x2, y2), we say P1 is **dictated by** P2 if x2 > x1 **and** y2 > y1 (both conditions have to be satisfied), otherwise P1 is not dictated by P2.

   For example, if P1 = (1, 3) and P2 = (2, 6), then P1 is dictated by P2 (as both 2 > 1 and 6 > 3). However, if P1 = (4, 3) and P2 = (5, 1), then P1 is not dictated by P2 (as 5 > 4 but 1 < 3, so both conditions are not satisfied). Similarly, if P1 = (3, 5) and P2 = (1, 2), then P2 is dictated by P1 (as both 3 > 1 and 5 > 2).

   You will use 2 arrays X and Y. X will store the x-coordinates of the points and Y will store the Y-coordinates of the points. So the points will be (X[0], Y[0]), (X[1], Y[1]), (X[2], Y[2]),…and so on.

   Write a C program that does the following:
   - Declares two arrays X and Y to store maximum 100 integers each
   - Reads in a positive integer n (n < 100)
   - Reads in the coordinates of n points. **In a single loop**, read in the x-coordinates of the points in X and y-coordinates of the points in Y (i.e., each iteration of the loop will first read the x-coordinate of a point, and then read the y-coordinate of the same point, and then go to next iteration of the loop to read the x and y-coordinates of the next point, and so on). The coordinates are all integers. **You must enter the inputs exactly in this order.**
   - Print the points read nicely. All the points should appear in a single line.
   - Find and print all the points that are **NOT dictated by any other point** in the set of points read.
   - Delete all the points that are **NOT dictated by any other point** in the set of points read from X and Y arrays. Print the points remaining from the X and Y arrays after deletion.

   See example below for format of the output to be printed. You can use additional arrays if you want.

Example:

Points entered: (3, 7), (5, 10), (2, 4), (6, 13), (7, 12)

Your program should print

The points entered are (3, 7), (5, 10), (2, 4), (6, 13), (7, 12)
The non-dictated points are (6, 13), (7, 12)
Remaining points are (3, 7), (5, 10), (2, 4)