Online Assignment 1: List (30/11/2021 Tuesday-Afternoon)

You need to write new List implementations (both Arr based and LL based) that have all the functions of the List ADT as per the specification of your offline assignment (i.e., Table 1 of the relevant document) plus the following functions (c.f. Table 1A). You need to use your Arr based and LL based implementations to implement the functions defined in Table 1A to write your new implementations without changing any existing method.

Table 1A: Extended functions

Fn #	Function	Param.	Ret.	After functions execution	Comment
	[before function execution]			<20, 23 | 12, 15, 1>	The vertical bar indicates the current position. Here the current position is 2 (first position is 0) and corresponds to the element 12.
14	swap(pos)	0		<12, 23 | 20, 15, 1>	Swap the current element with the element in the position given by pos.
15	findMaxUptoCur()		23	<12, 23 | 20, 15, 1>	Returns the 'maximum' element from the beginning of the list upto including the current element. For a composite element

					maximum need to be defined as per need.
16	findMinUptoCur()		12	<12, 23 | 20, 15, 1>	Returns the 'minimum' element from the beginning of the list upto including the current element. For a composite element minimum need to be defined as per need.

Merit List (ML)

Now create a list of student information where each element contains a tuple <x, y>, where x is a student no. and y is the percentage of marks of student x. Now, given such a list you need to be able to prepare a merit list of students according to their percentage of marks (assume integer for simplicity). You will have to manipulate the list using the above functions (and the original functions in your offline assignment Table 1)

Input: You will have a number of lines containing a couple of integers, space separated, representing the tuple <x,y>. The end of input is marked by two 0s.

Output: You have to output in each line two integers x, y (space separated) where x and y are as defined above. The first line will contain the student having the highest marks, the second line will contain the student having the second highest marks and so on.

Example Input:

111 90

222 40

333 93

313 33

0 0

Example Output:

333 93

111 90

222 40

313 33

Submission Guidelines:

1. **Create a directory with your 7-digit student id as its name**
2. **You need to create separate files for the Arr implementation code (e.g. Arr1.cpp/Arr1.py), LL implementation code (e.g. LL1.cpp/LL1.py) putting common codes in another file. You will also include your previous implementations as separate files. Create a separate file for the main function implementing the ML part (e.g., ML.cpp/ML.py). No built in data structures or sorting algorithm can be used.**
3. **Put all the source files only into the directory created in step 1. Also create a readme.txt file briefly explaining the main purpose of the source files.**
4. **Zip the directory (compress in .zip format. Any other format like .rar, .7z etc. is not acceptable)**
5. **Upload the .zip file on Moodle in the designated assignment submission link. For example, if your student id is 1905xxx, create a directory named 1905xxx. Put only your source files (.c, .cpp, .java, .h, etc.) into 1905xxx. Compress the directory 1905xxx into 1905xxx.zip and upload the 1905xxx.zip on Moodle.**

Failure to follow the above-mentioned submission guideline may result in upto 10% penalty.