Notes:

* You are required to upload your in-class implementations of problems 1 to canvas. This is due by 9:50 AM today.
* You are required to turn in a written report (Word or PDF file) for the homework part (problem 2) of the lab and upload implementations to canvas. These are due by 8:00 AM, October 11, 2016.
* Homework report must follow the guidelines provided earlier.

**Objectives:**

* Learn to come up with novel sorting ideas
* Learn to choose the best sorting algorithm for a given problem.

**Problems**

1. Consider an algorithm that sorts an array of *n* elements by finding the smallest and largest elements and then exchanges those elements with the elements in the first and last positions in the array. Then the size of the array is reduced by two elements after excluding the two elements that are already in the proper positions, and the process is repeated on the remaining part of the array until the entire array is sorted.

Write a code to implement the above algorithm. Write a driver program to show the novel sorting algorithm works correctly.

1. Consider an application that logs transactions. The log includes the location where a transaction originated and the time the transaction occurred and this information is stored such that they are ordered by the time of the transaction (see sample input below). You are required to sort this log by location while preserving the order of the time field (see sample output below). Implement an algorithm of your choice to sort this array based on the location while preserving the order of the time field. Test your algorithm with the input file provided in Canvas named “NovelSortInput.txt”. Explain in your report why the sorting algorithm you chose is the best for the job.

See a sample output below:

|  |  |
| --- | --- |
| Sample Input | Sample Output |
|  |  |