**MAT 2540: Data Structures and Algorithms II**

|  |  |  |
| --- | --- | --- |
| **PROJECT 1**  |  |  |

The goal of this project is to make you become familiar with the basic implemented in a computer programming language, like C++. For this assignment, please first visit the links <http://cpp.sh/8vbls> for the **Maximum** and <http://cpp.sh/9f5o> **BubbleSort Programs**, writtenin C++.

**Assignments:**

1. Create a **Minimum Program** in C++**,** by modifying the **Maximum Program** <http://cpp.sh/8vbls> which we discussed in class. If you have any questions about how the program works, feel free to post it on Piazza.
2. Recall the definition of **selection sort algorithm**: We start with an integer array of ***n*** elements. In first step, we choose the maximum of the **first n elements,** and put it at the end of the list (to do this, you may interchange the nth element in the array with the maximum element). Then, we find the maximum of the remaining **n-1** elements, and put it at the **n-1st** location in the array, right before the nth element. And so on. The process will stop after **n-1** steps. Write a **pseudocode** that describes this algorithm. You can look at the pseudocode for **bubblesort** and **insertionsort** (see the slides for Chapter 3 on BlackBoard) as a guidance.
3. By utilizing the **maximum function** above, create a code for **SelectionSort** in C++. You should use the **bubblesort program** http://cpp.sh/9f5o as the template to modify. Include the pseudocode you wrote in Step 2 at the beginning of the program as a comment. As in **BubbleSort** program, your program must first ask the user about how many integers they want to enter. Then record these integers in an array. Apply SelectionSort, and finally print the list. Make sure to delete all unnecessary line of codes after you are finished. The end result should also be user-friendly, that is, make sure your output is readable and understandable by any person who doesn’t know any programming language.
4. Save the **minimum** program and the **selectionsort** program on cpp.sh. Create a short URL for each program, by clicking the “Get URL” button.
5. Finally, think of **other uses** of the programs you just created. Implement **one example** on C++, and **create a short URL** for your work.

**Submit the following by e-mail by the due date:**

1. Three cpp.sh links containing the codes for **Minimum**, **SelectionSort**, and the **extra program** in Step 5 above.
2. **A short paragraph** that explains how you extended the code in Step 5 to accomplish other tasks.

**Grading Scheme:**

**70% for the accuracy**: Your code is expected to do everything what is asked in the assignment. It must be error-free. Especially if you are working on this assignment in pairs...

**20% for the clarity**: You are expected to write comments within your code that explain the role of each of the line you add to the original code. Pay attention to spacing and indentation. Name your variables and functions appropriately.

**10% for additional work**: Think of other possible uses of the code you've written, and extend it to make it more useful.