

## CS111 – Introduction to C Programming

(Draft)

Fall 2025

Programming Assignment #6 (3%)

Due: Friday October 24, 2025 (23:59)

1D Array Applications: Statistics and Sorting

### Objectives

In this programming assignment, you will use C 1D arrays to compute the statistics (max, min, and average) of a list of numbers. As well, you will also write a C program that implements a simple algorithm to rearrange a list of numbers into the ascending order, a process known as *sorting*.

### Part I: Calculating the minimum, maximum, and average of a list of numbers

In this part of the assignment, you will write a program that finds the minimum, maximum and average of a list of numbers, which can be quite useful, for example, after an exam when the instructor provides the statistical results of the exam to students. The numbers are integers and stored in an array whose size is fixed and defined by a constant called **N**, in the same way as in the example on p. 164 of the text. Your program should prompt the user to input **N** numbers and read them into an array with a **for loop**. In the second **for loop**, your program should calculate the minimum, maximum and the average of the **N** numbers and print out the results after exiting the second **for loop**. (You could do everything in one **for loop**, but for simplicity, implement this program with two **for loops**.) Note that the average of **N** integers is not necessarily an integer, and so you will have to print out the average with the **%f** format specifier. Read p. 147 of the text to learn how to use the casting operation in C to generate a **float** from an integer division. Submit the program as **pa6p1**.

### Part II: Bubble Sorting

Sorting is an important operation used in numerous computer algorithms. It refers to the process of rearranging a set of numbers into ascending (or descending) order. Many algorithms exist to solve the sorting problem but *bubble sort* is perhaps the easiest to understand, although it is not the most efficient. The pseudocode below defines how bubble sort works.

```
i = N;
sorted = false;
while ((i > 1) && (!sorted)) {
    sorted = true;
    for(j=1; j<i; j++) {
        if(a[j-1] > a[j]) {
            temp = a[j-1];
            a[j-1] = a[j];
            a[j] = temp;
            sorted = false;
        }
    }
    i--;
}
```

Based on the pseudocode above, write a program that implements bubble sort. As in Part I, your program should prompt the user to input **N** integer values and store them in an integer array. Then the program should proceed to sort the **N** numbers into the ascending (increasing) order by following the algorithm in the pseudocode above. Finally, the program should print out the sorted array of numbers. Once your program works, make sure that it is properly documented and name it as **pa6p2.c**.

### **Marking and Assignment Submission**

This programming assignment is worth a total of 3% with 1.5% for each part. Submit **pa6p1.c** and **pa6p2.c** as well as the zipped folder of the two files, called **pa6.zip**, as three separate files through [Blackboard](#). Be prepared to explain to a TA in the next lab session about your solution to this assignment.

**Notes:** Please work on the homework independently. The university has a zero-tolerance policy on plagiarism. Regarding the use of AI to assist you in completing assignments, please refer to “AI Policy” on course Blackboard.