Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth

ECE160: Foundations of Computer Engineering I (Spring 2023)
Instructor: Dr. Liudong Xing

LAB # 11

(Relevant Lecture: #22-24)

Monday, April 10 (L1) and Wednesday, April 12 (L2)

OBJECTIVES

- To continue practicing the use of arrays
- To practice the bubble and selection sorting

SUBMISSION REQUIREMENT

1. Please follow "Submission Guidelines" in the lab section of the course website to submit your solution (program files) to the class M: drive by 5pm, Wednesday, April 12
2. Suggested format for naming your solution files: lab#-your last name-p#.cpp
   For example: lab11-xing-p1.cpp for problem 1; lab11-xing-p2.cpp for problem 2; …

EXERCISES

1. Write a program that reads 8 integers into a 1-D array from the keyboard. Then print out the array elements. Then **swap** the first element and the seventh element (Refer to the example on Slide 6 in Lecture#22). Finally print out the array elements after the swapping.

   Testing: if you input 1 3 5 7 9 2 4 6 from the keyboard, then the output on the screen should be:

   ```
   The array elements before swapping: 1 3 5 7 9 2 4 6
   The array elements after swapping: 4 3 5 7 9 2 1 6
   ```

   Note: *the output in the test cases is given as an example. You may use a different format, for example, each number may be output on a different line.*

2. Modify the program in Exercise (4) on Slide 23 of Lecture #22 (refer to the corresponding solution file for the complete program) so that the program reads a 2-D array with 4 rows and 3 columns from the keyboard. Print out all the four elements of **the second column,** and print out the one element on **the third row and the third column** (one number per line).

   Testing: if you input 1 2 3 4 5 6 7 8 9 10 11 12 from the keyboard, which corresponds to the 2-D array with the following 4 rows and 3 columns

   ```
   1     2     3
   4     5     6
   7     8     9
   10    11    12
   ```

Then the output on the screen should be:

```
The elements of the second column are:
2
5
8
11
The element on the third row and the third column is:
9
```

3. Modify the array size to 10 in the **bubble sort** program discussed in *Lecture#24, Slide 15* (Refer to the corresponding solution file for the complete program) so that it can sort the 10 elements of the array into non-decreasing order (i.e., from the minimum value to maximum value).

   Testing: run it using the 24 34 12 7 3 88 90 7 2 63 the output should be

   ```
   2  3  7  7  12  24  34  63  88  90
   ```

4. Modify the **bubble sort** program in Exercise 3 so that it can sort the 10 elements of the array into non-increasing order (i.e., from the maximum value to minimum value).

   Testing: run it using the 24 34 12 7 3 88 90 7 2 63 the output should be

   ```
   99  88  63  34  24  12  7  7  3  2
   ```

5. Modify the **selection sort** program discussed in *Lecture#24, Slide 12* (Refer to the corresponding solution file for the complete program) so that it can sort the elements of the array into non-increasing order (i.e., from the maximum value to minimum value) and the sorted array elements are output to a file named sortedArray.txt using fprintf() (instead of printing the sorted array elements on the screen).

   Testing: run the modified program using the input: 23 34 12 7 3 12 and check the file sortedArray.txt on your disk, which should contain

   ```
   34  23  12  12  7  3
   ```

   Review your Lab#10 exercises and Lecture#20 about how to write output to a file using fprintf().