EECE 2160 – Embedded Design: Enabling Robotics

Homework #5

Assigned: Thurs., Oct. 31, 2024. Due Sat., Nov. 9, at 11:59pm on Canvas

Two problems, 100 points

Problem 1 (50 points) Sorting an Array of Integers

Write a C++ program that generates an array of 10 random integer numbers between 0 and 99, prints the original array, sorts the array in ascending order, and prints the sorted array. Here are some suggestions on how to proceed:

1. You will write and run this program on a COE Lab Linux Terminal. See the Unix/Linux Tutorial Lecture slides for directions on how to open the terminal. If you are working from home then you must use the VPN (GlobalProtect) software to establish a secure link to the COE server. You can download VPN (GlobalProtect) from here:

https://coe.northeastern.edu/computer/student-resources/engineering-software/#vmware

Be sure to watch the video on installation and set-up.

2. You can start by writing a function named PrintArray with the following prototype:

```
void PrintArray(int v[], int size)
{
    ...
}
```

This function takes an array v of integers as the first argument, and its size in number of elements as the second argument. The function traverses the array with a loop and prints all elements between 0 and size – 1.

3. Write another function named RandomArray with the following prototype:

```
void RandomArray(int v[], int size)
{
    ...
}
```

This function should initialize size elements of array v to random values between 0 and 99. You can use function rand() to generate random numbers. **You can obtain information on how to use this function from the manual pages with shell command man 3 rand**. You can limit the range of the generated random number with a modulo operation (operator % in C++).

4. Write a function named SortArray with the following prototype:

```
void SortArray(int v[], int size)
{
      ...
}
```

As its name suggests, this function sorts array v composed of size integer elements in ascending order. Use the *selection sort* algorithm to sort the array. This algorithm is based on finding the minimum element on the right of the array and placing it on the left in an iterative manner. In each iteration, the array will be split in two parts: one that is sorted on the left, and one with the remaining unsorted elements on the right. As the algorithm progresses, the sorted part grows, and the unsorted part shrinks. You can find more information about selection sort in the following Wikipedia page, which includes a basic implementation of the algorithm and an animation illustrating its behavior:

https://en.wikipedia.org/wiki/Selection_sort

5. Finally, write a main() program that instantiates an array of 10 elements and invokes the previous functions in the appropriate order.

Hand-in:

1. (30 points) The source code of your program with comments explaining how the program works.
   a. (15 points) General comments about how the program works.
   b. (*15* points) Comments at major parts of the program, not necessarily line by line.
2. (20 points) The output of your program as shown on the terminal screen. In MobaTerm click on terminal in the tool bar, select “save terminal text” and then select a directory to hold the file. Save and submit the output as a text or .docx file for submission. Alternatively, you can save the screen as a .pdf file if you select “print terminal text” and then select Adobe pdf.

Problem 2 (50 points)

Write a C++ program that reads in 10 random strings from the user, then sorts them in alphabetical order, and prints them sorted. You can start with your integer sorting program above and modify it accordingly.

Hand-in:

1. (30 points) The source code of your program with comments explaining how the program works.
   a. (15 points) General comments about how the program works.
   b. (15 points) Comments at major parts of the program, not necessarily line by line.
2. (20 points) The output of your program as shown on the terminal screen. In MobaTerm click on terminal in the tool bar, select “save terminal text” and then select a directory to hold the file. Save and submit the output as a text or .docx file for submission. Alternatively, you can save the screen as a .pdf file if you select “print terminal text” and then select Adobe pdf.