Programming Exercise 1

EE312 Fall 2013

To complete during recitation, the week of 9/2

TWO POINTS

You will be provided with a simple program that contains at least one error. Your challenge during this exercise is to identify the error(s) in the program and repair them. To receive credit for the exercise, you must complete the steps below during the time allotted by your recitation TA. The TA will be available to guide you, answer questions and will determine whether you've successfully completely the exercise.

STEP1: Acquaint yourself with the design.
The program you've received attempts to sort an array of integers. The algorithm used is called "selection sort". The algorithm is simple enough. We know that the smallest element in the array should be first, so we start by finding the smallest element and then we put that element first. We find the smallest element using our standard min idiom (implemented by the function called "min") and we put the smallest element first by swapping the minimum element with the element at the start of the array (using the swap function). We then repeat this process for the 2nd smallest element and so on, until we've sorted the array.

Draw a diagram that represents the state of this program. Your diagram should illustrate the state of the whole program, including local variables for each of the functions. Using your diagram, execute the program (by hand, not on the computer) up through the point where the smallest element is placed at the start of the array. Have your TA confirm that your diagram is reasonably complete and accurate. (the TA must check your diagram for you to receive credit for this exercise).

Step 2: Set a breakpoint and launch the debugger.

Set a breakpoint inside the "min" function at the start of the function. Launch the Visual Studio debugger. While the debugger is running examine the state of all three stack frames, main, sort and min. Confirm that you can look at the values of local variables in each of the three frames.

Step 3: Single-step the program through min.

Using the single-step function in the debugger, step through the first invocation of min. Confirm that the value returned is the correct address. Update the diagram you produced to show the actual addresses where the array x[] is stored. Confirm that the pointers used by min are accessing the correct memory locations.

Step 4: continue the program and repeat Step 3

Hit the continue program to resume execution. The debugger should stop in min when it hits the breakpoint. Confirm that the value of “k” in the sort function is equal to 1 (i.e., we’re looking for the second-smallest value). Step through min. Refer to the diagram you produced during step 3. Is the correct address returned? Why not?

Step 5: continue debugging the program and correct any errors you discover

You’re on your own here. Good luck.

Step 6: design at least two additional tests for the sort algorithm.

I’d like for each student to develop at least two tests. Then I’d like the group to discuss what tests they picked and why. Is there a set of three or four tests from the whole class that look to be a good way to confirm that the sort function works properly? As long as there’s some thought put into the question of “how to test”, I’m happy. There’s no “right answer” to this part of the exercise.