CS16, 09F, UCSB

H07: (Selection sort, initial C code) Total Points: 50 (printable PDF)

Accepted: on paper, **IN LECTURE** on 09/30 **10/30**
But, bring it to LAB with you on 09/29 or 09/30 **10/29 or 10/30**

(You may make this up ONLY for excused absence from lab due to illness or excused absences arranged in advance, e.g. attending a conference related to academic or school sponsored student activities—and you may only make it up during Conrad's office hours, or by appointment, within one week of 10/30.)

DO THIS AS PREPARATION FOR YOUR DISCUSSION SECTION (lab) on 10/29 or 10/30
You may do this individually, or together with your pair programming partner—but be sure that you each hand this one in separately.

Name : (3 pts)______________________________ Umail address (3 pts) _____________________

Section (4 pts) Circle one: Thu 8am Fri 10am Fri 11am Fri noon

Name of your pair programming partner(s), if you work together:

About this assignment: In lecture on Monday 10/26 and Wednesday 10/28, we went over the main ideas of selection sort. Your job here is to begin to turn those ideas into C code.

For ***this*** assignment, it is not necessary that your code be perfect. We are grading you on whether you have made a good faith effort to try to write code that will work, and have prepared yourself for lab on Thursday/Friday.

If you are confident in your solution, that's great. If you are lost, please come see your instructor or a TA for help.
Conrad has office hours today 10/28, 4:15-5:15, and Harry has office hours 1-2pm on Thursday 10/29.

---

1. (20 pts) First, write a function called indexOfMax that computes the index of the largest element in an array of ints.

   The function should take these parameters: an array (call it `a`) and the size of the array (call it `n`, or `size`)
   It should return the index of the largest value in the array (a number between `0` and `n-1`)
   (In case of ties, return the index of *any one* of the values that is "tied for largest".)

Please turn over for more problems

Continued from other side

2. (20 pts) Write a function called selectionSort that sorts an array.

   The parameters to your function selectionSort will be the same as the parameters to indexOfMax.
   The function will be void—it operates by having a "side effect" on the array that is passed in (i.e. it sorts it.)

   Use the function that you defined in part one. You can use a function call like this one to find the location of the maximum value among all the first "howMany" elements of a.

   ```
   maxLoc = indexOfMax(a, howMany);
   ```

   With that tool, you should be able to write a function that performs selection sort by doing this repeatedly, on smaller and smaller parts of the array (all the elements, then the first n-1 elements, then the first n-2 elements, etc. until you are down to just 2 elements.) Each time, you want to swap the largest value with the element at the "end" of that part of the array.

   Remember that to swap two values, you need code like this:

   ```
   temp = a[i];
   a[i] = a[j];
   a[j] = a[i];
   ```

End of H07