

Project 2 (Sorting Competition): EE368 Spring 2015

Assigned Date: February 24, 2015
Due Date: 10:00am March 13, 2015

Note: Satisfactory completion of this project satisfies an ABET curriculum outcome for this class (outcome 5). You must earn a score of 60% or higher on this project in order to receive a passing grade for this class.

Write a sorting function to sort integer arrays (integers can be positive, negative or 0, and can possibly occur with repetition). The prototype for the function must be as follows:

```
void sort(int list[], int n);
```

where list is a pointer to an array of integers, and n is the number of integers in list[].

Grading

The test program will call your sort() function and pass various lists of varying sizes and degrees of sortedness. The total time for sorting all the lists will be tallied for each student. Grades will be assigned according to your total sorting time and, in particular, will be based on the ranking of the execution time of your sorting algorithm. The rank-based scoring will be done as follows:

The two students with the fastest-running programs will receive the highest score of 100. The next two students (with the third and the fourth-fastest running programs) will score 99, the next two students will receive a score of 98, and so on all the way down to the score of 60. Note, in this manner 82 students will score ≥ 60 .

Score for the remaining students will be assigned at the individual level, starting with the score of 59 and counting downward. In order to pass the ABET outcome, students with a score < 60 can resubmit their program which must meet a minimum execution time requirement (to be specified later on). Resubmission period will be of 5 days after the release of Project 2 grades. **However, the original grade of a student in this group will not change.**

Compiling

The scoring guidelines above only apply to programs that work properly. If your program will not compile or fails to sort the list correctly, you will receive a 0 grade. Be sure that your program compiles successfully with the command

```
gcc -Wall project2.c project2_test.c
```

on one of the Linux machines. The file project2_test.c contains test routines and a main() function.

Additional Notes

- Do not write any output to STDOUT or STDERR in the program you submit for grading.
- You can use multiple (combination) of sorting functions if you wish.
- You may use the Unix C functions `clock()` or `times()`, or the Unix command `time` to determine your program's running time for your own purposes. Note that the granularity of the timing functions can be large and it may be necessary to sort arrays of several thousand elements before they return nonzero values for elapsed time.
- Submit your project using the `turnin` command from the directory containing your `proj2.c` file:

```
turnin -c ee368ta -p proj2_s15 project2.c
```