

# Homework 4

## Pointers and the Funny Things You Can Do With Them

Assigned: Monday, September 28, 1:30PM

**Due: Monday, October 5, 1:30PM (Hard Deadline)**

The purpose of this assignment is to act as a refresher on how to work with pointers, especially function pointers, in C. Additionally, it refreshes the new concept of weak references.

### Grading

#### *Part I: Software*

- 20 points – Everything works.
- 10 points – You put effort in but it doesn't quite work.
- 0 points – Little to no effort / nothing works / nothing submitted.

#### *Part II: Follow-up Questions*

- 10 points – [Follow-up questions](#)

*Honor Code Reminder:* In the interest of making this easy for you to implement, test, and submit this is probably one of the easiest assignments to cheat on ever. Please don't. Resist the temptation to “glance” at another solution for help because you got stuck. There just isn't enough code that a “glance” will do anything but give you the solution. This assignment is deliberately worth very few points; if you didn't do it, just don't submit anything [why risk cheating?]. The point of this assignment is to make your life easier later this semester, don't cheat yourself.

### Assignment

Grab a copy of the code from <http://github.com/eecs373-f15/373-f15-function-pointers>. I recommend forking this repository to do your work.

First things first, type `make main && ./main` and look at the output. Look over `main.c` and `sort.c` to try to understand what the current code is doing. Specifically, how does `main` call the different sort functions in `sort.c`? How does `main` know how many sort functions there are? What is the purpose of the `compare` function?

**Task 1:** We will add a third sort algorithm. Instead of implementing it ourselves, we will use the built-in `qsort` from the C standard library. For details on `qsort`, type `man qsort`. The `qsort` type signature doesn't quite match our `sorting_fn` type signature, you will need to write a wrapper function.

Do your work for Task 1 in the `sort.c` file. You may only edit `sort.c` for this task.

When you are done with this task, type `make check_main` to check your work.

**Task 2:** Next we will modify the sort functions to reverse the order they are sorting in. We will do this *without* modifying `sort.c`. Type `make reverse && ./reverse`. Currently this will behave the same as `main`. The difference between `main` and `reverse` is that `reverse` also links in `reverse_sort.c`. Add a function to `reverse_sort.c` so that numbers are now sorted in descending order. You may find it useful to consult Lab 4 for a refresher on weak references.

Do your work for Task 2 in the `sort.c` file. You may only edit `reverse_sort.c` for this task.

When you are done with this task, type `make check` to check your work.

---

## Notes

I have tested this on my (Linux) desktop, the CAEN Linux machines, and my personal mac. If you have any issues building / running the initial code, please file an issue on GitHub (the “Issues” button at the right side of the page).

## Submission

### *Part I: Software*

Your `sort.c` and `reverse_sort.c` implementations will be automatically graded. To submit, go to <https://docs.google.com/spreadsheets/d/1AGWtXCvt7iAtThuRlz3XovYnD2dfjzQA6idmVA6sM80/edit?usp=sharing> and add your unique name and a link to download your `sort.c` and `reverse_sort.c`.

A script will automatically download, compile, and grade your `sort.c` and `reverse_sort.c` implementations. This script will run automatically at 1:35PM on Monday, October 5th (to account for clock differences, don't rely on this). **No late submissions will be accepted.** A test run of the script will run at 1:35PM on Sunday, October 4th. In both cases, the script will immediately email the results.

### *Part II: Follow-up Questions*

We will be trying Gradescope for this part of the assignment. Download the follow-up questions [here](#). This is an editable PDF, so you should be able to simply type in your responses (be sure to save a copy with your responses!). If that doesn't work, you can simply print and scan a copy of the assignment.

You should all have been added to EECS 373 on Gradescope. If you are having difficulties, post on Piazza.