



# Computer Science 381

## Programming Unix in C

The College of Saint Rose  
Winter Immersion 2016

## Lab 4: Introduction to Pointers and Arrays

Due: Tuesday, December 29, 2015

In this lab you will start to learn about pointers and the closely related topics (at least for C programmers) of arrays and strings. Even if you never write a C program after this course, understanding how C programs manage pointers will improve your programming in nearly any language.

Recall that you will also be finishing up the programming assignment from last week's lab, so there is not much additional programming here.

---

### Read About It

Read the first 4 sections of Chapter 5 in K&R. It's only 11 pages, but it's a pretty intense 11 pages.

#### ? Question 1:

In your own words in a sentence or two, explain what happens when you execute a statement like `p = &c;` at the top of page 94. (1 point)

#### Practice Program:

Write a C program called `swapper.c` that demonstrates that the `swap` function in the middle of p. 95 does not perform a swap of the values passed in (from the perspective of the caller), but the `swap` function at the top of p. 96 does. Include tests that attempt to swap regular `int` variables as well as array entries. (7 points)

#### ? Question 2:

Explain briefly, again in your own words, why the first `swap` function cannot work as intended. (2 points)

---

### A Pointer/Array Example

#### See Example:

`/home/cs381/examples/isort`

The above example includes two C functions that each perform an insertion sort on an array of `int`. You can switch between the two implementations of the sorting function to verify that both do work.

**? Question 3:**

Draw memory diagrams (like those on p. 96, but show all variables in existence in both `main` and the `sort` function) to show the state of memory during the second iteration of the `for` loop right before the `while` loop starts in the `isort` example. Include diagrams for an execution using the `isort` function as shown in the example and when that call is replaced with a call to `isortptr`. (6 points)

** Practice Program:**

Add a function to `isort.c` to perform a pointer-based array printing (along the lines of `isortptr`) called `print_array_ptr`. (4 points)

**Submission**

Please submit all required files as email attachments to [terescoj@strose.edu](mailto:terescoj@strose.edu) by Tuesday, December 29, 2015. Be sure to check that you have used the correct file names and that your submission matches all of the submission guidelines listed on the course home page. In order to email your files, you will need to transfer them from mogul to the computer from which you wish to send the email. There are a number of options, including the `sftp` command from the Mac command line.

**Grading**

This lab is graded out of 50 points, including the programming assignment described in the previous assignment.

Grading Breakdown	
Lab questions	9 points
Practice program <code>swapper.c</code> correctness	7 points
Practice program <code>isort.c</code> pointer-based printing	4 points
<code>extremes.c</code> correctness	20 points
<code>extremes.c</code> design	3 point
<code>extremes.c</code> documentation	3 points
<code>extremes.c</code> style	3 point
<code>extremes.c</code> efficiency	1 point
<b>Total</b>	<b>50</b>