Todd Sipe
CS 241
Lab 4 write up

	Total Time For 2d Array	Total Time For Pointer Array
1,000 iterations	0.001845 seconds	0.000728 seconds
10,000 iterations	0.018097 seconds	0.005228 seconds
100,000 iterations	0.167581 seconds	0.032429 seconds
1,000,000 iterations	1.668633 seconds	0.310917 seconds
10,000,000 iterations	17.318535 seconds	3.042397 seconds
100,000,000 iterations	174.263404 seconds	30.308323 seconds

Although the process and for loop are almost identical in the end, especially with all of the heavy lifting of creating the rest of the code out of the way, (thanks!) I would say that the more difficult version to code up was the one that was referencing elements in an array of pointers. I was a bit foggy to start with exactly how to use this. I struggled for a while to figure out how to print the elements, and after searching for quite a while I found that:
puts(pointer_arr[0]);
Worked fine for printing out the element of the array. I had to do a bit of learning to figure out that it was actually much simpler than I thought at the approach. This ALWAYS inevitably leads me to a great deal of learning though, and I now feel vastly more comfortable with using pointers.

The more efficient way to handle strings is clearly with using pointers. The difference between three minutes and some change and thirty seconds is undeniable. That having been said, only if we breach the 10,000,000 iterations point do we need to think about the run time efficiency. I think that the reason for this is that the pointers just swap the pointer, which (I think) is only one character for reference for where the string is stored in memory. If we have to swap the entire string WITHOUT using pointers, we are swapping roughly 19 times more information, because we are swapping every character of the string.

The results were interesting and fun to observe. I have never used any program so far in three terms of CS classes that iterate through something millions of times, and so I have never seen the actual difference between run times for something like this in comparison. We learned about pointers in class, and I followed mostly how it worked (like I said I was rough on syntax until I jumped into it), and it was funny to me because on Father's Day I asked my father in law who is a professional in the field why pointers were even used or relevant, and he got VERY excited and was about to explain it to me, but never did, cuz there was fun stuff happening. Then we come into class and work on a project that literally shows me the power of pointers. It was extraordinarily fulfilling to not only learn the difference but to work it out by coding something myself.