CS 161
Intro to CS I

More Arrays vs. Structs
Chap. 6.1
Quiz #7

• Get into groups of 3-4.
• Draw the picture and pseudo code for creating an int ****p; that points to int i;
• Describe an example structure you might define.
• How would you return a struct from a function?
• How would you create an array of these structures?
Pointers Demo....

```cpp
#include <iostream>

int main() {
    int ****p, ***p1, **p2, *p3, i = 10;

    p = new int***;
    *p = new int**;
    ***p = new int*;
    ****p = &i;

    // Another way to make p point to i
    /*p = &p1;
    p1 = &p2;
    p2 = &p3;
    p3 = &i; */

    std::cout << ****p << std::endl;

    // delete ***p; // Don't do this because i is on stack
    delete **p;
    delete *p;
    delete p;

    return 0;
}
```

"pointers.cpp" 25L, 375C written 2,0-1 All 3
```cpp
#include <iostream>

struct doc_record {
    char name[10];
    int age;
    float weight;
};

void using_structs(doc_record []);

using std::cout;
using std::endl;

int main() {
    doc_record family[3];
    family[0].name[0] = 'A'; family[0].name[1] = 'K'; family[0].name[2] = 'M';
    family[0].name[3] = '\0';
    family[0].age = 5;
    family[0].weight = 37.5;
    using_structs(family);
    cout << "Name1: " << family[0].name << ", Name2: " << family[1].name << endl;
    return 0;
}

void using_structs(doc_record f[]) {
    f[1].name[0] = 'J';
    f[1].name[1] = 'P';
    f[1].name[2] = 'M';
    f[1].name[3] = '\0';
}
```
Why is it good to have an array of structs?

• What happens if you have two arrays with first names and last names, and you want to sort by first name?
• What happens if you put the first name and last name in a struct?
#include <iostream>
#include <cstring>
#include <stdlib.h>

#define NUM_PEOPLE 3

using std::cout;
using std::endl;
using std::cin;
using std::string;

struct friends_rec {
  string first;
  string last;
};

int cmp(const void* a, const void* b) {
  return strcmp(((friends_rec *)a)->last.c_str(), ((friends_rec *)b)->last.c_str());
}

int main() {
  friends_rec friends[NUM_PEOPLE];
  friends[0].first = "jennifer"; friends[0].last = "parham";
  friends[1].first = "austin"; friends[1].last = "mo";
  friends[2].first = "brian"; friends[2].last = "mocello";
  qsort(friends, NUM_PEOPLE, sizeof(friends_rec), cmp);
  for (int i = 0; i < NUM_PEOPLE; i++)
    cout << friends[i].first << " " << friends[i].last << endl;
  return 0;
}