CS 161
Intro to CS I

Revisit Recursion and Arrays
Odds & Ends

• How are you on Assignment #4?

• Revisit Static vs. Dynamic Arrays
```c
#include <iostream>

int main (int argc, char *argv[]) {
    float grades[NUM_STUDENTS][NUM_GRADES];
    int g[3]={0, 10, 20}; //Create a static array

    cout << "Static 1-d" << endl;
    cout << &g << endl; //This is where g lives
    cout << g << endl; //Contents of g, which is where g[0] lives
    cout << &g[0] << endl; //Where g[0] lives
    cout << &g[1] << endl; //Where g[1] lives, which is beside g[0]

    cout << "Static 2-d" << endl;
    cout << &grades << endl; //Where grades lives
    cout << grades << endl; //grades has where grades[0][0] lives
    cout << grades[0] << endl; //Array of const self-ref pointers
    cout << &grades[0][0] << endl; //Where grades[0][0] lives
    cout << &grades[0][1] << endl; //grades[0][1] lives beside grades[0][1]
    cout << "----------" << endl;
    //Next element in array of self-ref pointers points
    //to the next row in the array, which is why grades+1
    //takes you a whole stride/column length
    cout << grades[1] << endl; //Contains where grades[1][0] lives
    cout << &grades[1] << endl; //Lives there too because self-reference
    cout << &grades[0][2] << endl; //Where last element in 1st row lives
    cout << &grades[1][0] << endl; //Rows are consecutive in memory
    cout << &grades[1][1] << endl; //Elements in a row are consecutive

    // -- INSERT --
    return 0;
}
```