CS 161
Intro to CS I

2d arrays and Command-Line Arguments
Odds and Ends...

• Last week to demo Assignment 4
• Test next Wed., 11/22
Static 2-D arrays...

- Constant, self-ref
- Pointer to the row pointers
- Row of pointers
- Contiguous
- Fast

```c
int array[2][3];
```
Passing a 2-D Array (Static)

int main() {
    int array[5][5];
    ...
    pass_2darray(array);
    ...
}
void pass_2darray(int a[5][5]) {
    cout << "Array at zero: " << a[0][0] << endl;
}
OR
void pass_2darray(int a[][5]) {
    cout << "Array at zero: " << a[0][0] << endl;
}
```cpp
#include <iostream>

using namespace std;

void fun(int a[][3], int rows, int cols) {
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            a[i][j] = 10;
}

int main(int argc, char *argv[]) {
    // how do I create a 2-d array (3 x 3) on stack?
    int array[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    fun(array, 3, 3);
    // how do I print address of pointer to 1st row pointer?
    cout << &array << endl;

    // how do I print address of 1st row pointer?
    cout << array << endl;
    cout << &(array[0]) << endl;

    // how do I print address of the 1st element in 1st row?
    cout << array[0] << endl;
    cout << &(array[0][0]) << endl;

    // how do I print address of the 1st element in 2nd row?
    cout << array[1] << endl;
    cout << &(array[1][0]) << endl;

    // print contents of 2nd element in 2nd row
    cout << array[1][1] << endl;
    cout << *((array + 1) + 1) << endl; // works on dynamic/static
    cout << *((array + 1) + 3 + 1) << endl; // only works on static
}```