

EECE.2160: ECE Application Programming

Spring 2016

Lecture 20: Key Questions

March 11, 2016

1. **Example:** Complete the following program:

```
#include <stdio.h>
#define NRows 3      // # of rows
#define NCols 4      // # of columns
int main() {
    double x[NRows][NCols] =          // 2-D array
        { { 10, 2.5, 0, 1.5},
          {-2.3, -1.1, -0.2, 0},
          {10.5, -6.1, 23.4, -9.2} };

    int negCnt[NRows] = {0};          // Initialize entire row
                                        // count array to 0
    int i, j;                          // Row and column indices

    /* INSERT CODE HERE--Visit every element in array x and
       count the number of negative values in each row */

    // Now print the row counts
    for (i = 0; i < NRows; i++)
        printf("Row %d has %d negative values.\n",
               i, negCnt[i]);

    return 0;
}
```

2. Explain how to pass arrays to functions.

3. **Example:** Write a function for each of the following:

- Given an array of doubles (`arr`) and the # of elements in the array (`n`), find the average of all array elements

- Given an array of ints and the # of elements, find the largest element in the array

- 3

4. Explain the relationship between pointers and arrays.

5. Explain how 2-D arrays are passed to functions.

6. **Example:** Say we have a program that stores student exam scores in a 2-D array:
- Each row represents an individual student
 - Each column represents one of the 3 exams

Write functions to:

- Calculate the exam average for each student and store it in a 1-D array that is accessible in the main program
 - Assume all exams have equal weight
- Calculate the average for each exam and store it in a 1-D array that is accessible in the main program
- Each function takes the same arguments:
 - The 2-D array
 - The # of students in the class
 - The 1-D array that will be used to hold the averages

5 (cont.) Extra space to write functions

