**CS173: Intermediate Computer Science**

**Reading 8**

Name: \_\_\_\_\_\_\_\_\_Daniel Lee\_\_\_\_\_\_\_\_\_

Read the assigned pages below from our course textbook. Complete the responses to the questions in this document and then save as a docx or pdf file. Submit your work by the assigned deadline on the Canvas course page or in class. Responses may be neatly handwritten or typed. **Put your name at the top!**

Readings: From the course textbook please read Chapter 11.4-11.8.

You should come away with understanding:

* how to declare and use 2D arrays
* how 2D arrays are stored in memory
* how to pass 2D arrays as parameters to functions
* how to process 2D arrays

**1) Write a declaration below that creates a two-dimensional array of floats with 5 columns and 3 rows.**

float float2DArray[3][5];

**2) Explain why it is necessary when passing a two-dimensional array as a parameter to a function that you must specify the dimension of the columns but it is optional to specify the dimension of the rows.**

It is because in the computer’s memory, C++ stores two-dimensional arrays in row order and a function that receives an array’s base address must be able to determine that how many columns are in each row.

**3) Below is a declaration for a two-dimensional array of integers. Draw a map of how this array is stored in memory. How many bytes apart are** data[0][2] **and** data[2][0]**? You do not need to submit your drawing but you need to explain the calculations you did to arrive at your byte distance.**

data [2][0] is base + 48 bytes apart from the base and data [0][2] is base + 8 bytes apart from the base so the amount of bytes apart between data [2][0] and data [0][2] are 40 bytes.

int data [3][6];

·

·

·

data [1][5]

data [2][0]

data [0][1]

data [0][2]

data [0][3]

data [0][0]

base + 44

base + 48

base + 8

base + 12

base + 4

base

**4) Below is a function called miniMax. Write the code for this function. I suggest you test your function in a program and then copy/paste it here.**

*/\**

*This function finds and returns the smallest row-maximum value*

*in a 2D array.*

*PARAMETERS:*

*data: a 2D array of integers with ROWS rows and COLS columns.*

*ROWS and COLS are global const integers defined elsewhere*

*RETURN VALUE:*

*This function computes the maximum integer in each row.  It then returns the minimum value among all these row-maximums.*

*\*/*

#include <iostream>

using namespace std;

const int ROWS = 4;

const int COLS = 4;

int minimax(int data[ROWS][COLS]);

int main()

{

    int data[ROWS][COLS] = {

        {0, 1, 2, 3},

        {4, 5, 6, 7},

        {8, 9, 10, 11},

        {12, 13, 14, 15}};

    cout << minimax(data) << endl;

    return 0;

}

int minimax(int data[ROWS][COLS])

{

    int maxIntArray[ROWS];

    for (int i = 0; i < ROWS; i++)

    {

        int maxElts = data[i][0];

        for (int j = 1; j < COLS; j++)

        {

            if (data[i][j] > maxElts)

            {

                maxElts = data[i][j];

            }

        }

        maxIntArray[i] = maxElts;

    }

    int minValue = maxIntArray[0];

    for (int i = 1; i < ROWS; i++)

    {

        if (maxIntArray[i] < minValue)

        {

            minValue = maxIntArray[i];

        }

    }

    return minValue;

}