

## CS1050 – Prelab 5.1

Spring 2020

### Concepts to Practice

- Multi-dimensional arrays
- Passing arrays to functions
- Symbolic constants

### Description

For the prelab assignment, you should think about a group of students that are all taking classes. Conceptually, you might think of each class as a column in a table and you might think of each student as a row in that table. For example, consider this table:

Student	CS 1050	Math 1500	English 1000
Nelson Earle	72	68	62
Raj Mode	95	88	95
Roland Oruche	93	97	86
Dilruba Parvin	98	77	98
Sweta Praharaj	99	92	90
Jim Ries	47	32	27
Genevieve Saab	97	99	85
Jasmine Tan	85	95	91
Madison Williams	76	82	98

If this were Excel, it would be easy to get an average of all scores for CS 1050 (turns out it is 84.67 in this example). It would also be easy to get the average score for student Jim Ries (turns out to be 35.33 – this guy must not be attending class enough!).

Your job is to represent this information as a 2-dimensional array in C. You can leave out the headers and the student names for now (we don't want to embarrass "Jim Ries" too much), so this means you have 9 rows and 3 columns of integer data. Also, rather than hard-coding the number of rows and the number of columns in your array, you should use symbolic constants. That way, you can use these symbolic constants when calling the functions you will write as well.

Your program should print out the entire multi-dimensional array (matrix). Then, your program should calculate an average for each row (student) in your array and print that out. It should then calculate an average for each column (class) in your array and print that out.

Remember: No global variables. The two-dimensional array that contains all of the data should be declared and initialized in main. Main() should really only declare some variables and call the functions that you define (don't try to do all of the work in main()).

## Prototypes for the functions you should write

You may write any functions you wish to implement this program, in **addition** to the following functions. However, you **must** implement the following functions:

- **void Print2DArray(int array[][COLS], int rows);**  
This function should print all of the elements of the passed-in array in tabular form.
- **int PrintRowAverage(int array[][COLS], int whichrow);**  
This function should calculate the average for every element in the given row, and print this average.
- **int PrintColAverage(int array[][COLS], int whichcol);**  
This function should calculate the average for every element in the given column, and print this average.
- **int main(void)** – Of course, you need to write a main() 😊.

## Sample Output (bold indicates something entered by the user)

```
jimr@JimRHadesCanyon:~/CS1050/SP2020/labs/lab5.1$ compile prelab5.1.c  
jimr@JimRHadesCanyon:~/CS1050/SP2020/labs/lab5.1$ ./a.out
```

Here's the matrix:

```
Row 0: 72 68 62  
Row 1: 95 88 95  
Row 2: 93 97 86  
Row 3: 98 77 98  
Row 4: 99 92 90  
Row 5: 47 32 27  
Row 6: 97 99 85  
Row 7: 85 95 91  
Row 8: 76 82 98
```

Here are the row averages:

```
Average of row 0 = 67.333333  
Average of row 1 = 92.666667  
Average of row 2 = 92.000000  
Average of row 3 = 91.000000  
Average of row 4 = 93.666667  
Average of row 5 = 35.333333  
Average of row 6 = 93.666667  
Average of row 7 = 90.333333  
Average of row 8 = 85.333333
```

Here are the column averages:

```
Average of col 0 84.666667  
Average of col 1 81.111111  
Average of col 2 81.333333
```