This assignment gives you a chance to work with a two-dimensional array in C. A tar file has been provided in hw03.tar.gz. Download and un-tar this file with the command “tar -xvzf hw03.tar.gz” to create the hw03 sub-directory that has three files:

- matMult.c
- mat4x6_leftcol.txt
- Makefile

First, check out matMult.c. This program has three global integer matrices called matA, matB, and matC. These matrices are global because we want to update them with C functions, and we haven’t learned how to code a function that can update its own parameters, or return multiple values. To accommodate different sized matrices, I have chosen to make matA, matB, and matC relatively large. In the program, we will not use the entire matrix, but just the upper left corner of the matrix. The basic idea of the program is to provide a way to set the values of matA and matB, and then calculate the value of matC using matrix multiplication, where matC = matA x matB.

The program provides three ways of setting matA and three ways of setting matB. In matMult.c, there is the declaration of a function called setA_ID, another function called setA_Rand, and a function called setA_Stdin. The setA_Rand and setA_Stdin functions are already coded for you to give you examples of how these functions should work. You will need to code the setA_ID code, which show fill in the upper left rows and cols with an identity matrix… zeroes in every element of the portion of matA specified except those elements where the row and column index are the same. When the row and column index are the same, the element should have a value of 1. Note that the setA and setB functions should invoke the printMatrix function to show what has been set.

You will also need to code setB_ID, setB_Rand, and setB_Stdin functions that update the values of the specified upper left corners of matB. These functions should invoke printMatrix with a name of “B”.

There is also a function declaration called setC_Prod. You need to code this function as well. This function should calculate the value of matA x matB using matrix multiplication, and put the result in matC, and then print out the value of matC using the name “AxB”. There is a good description of matrix multiplication in Wikipedia: Matrix multiplication. Note that for matrix multiplication to work, the number of columns of the portion of the matA matrix we are using must equal the number of rows of the portion of the matB matrix; and portion of the result in matC will have the same number of rows as the rows portion of matA, and the same number of columns as the cols portion of matB. Therefore, the parameters to setC_Prod consist of ra – the number of rows of the matA matrix we are using, carb - the number of columns of the matA matrix we are using AND the number of rows of the matB matrix we are using, and cb – the number of columns of the matB matrix we are using.

Once you have coded these functions, you are ready to start testing. The matMult program is coded to take five command line arguments. First, a single letter, “I”, “R”, or “S”, which tells the program whether to use setA_ID, setA_Rand, or setA_Stdin to create the matA matrix. The second and third argument are the number of rows and columns to use in the matA matrix. The fourth argument is again a single letter, “I”, “R”, or “S” which tells the program how to fill in the matB matrix. The fifth argument is the number of columns to use in the matB matrix. (The program assumes that the number of rows to use in the matB matrix matches the number of columns in the matA matrix.)

Look in the Makefile for a couple of examples of how to invoke matMult. You will want to test your program under some other conditions as well. For instance, what happens if you make matA random, and matB an identity matrix of different shape? What happens if you make matA random, and fill in matB with the values in mat4x6_leftcol.txt?

When you are done testing, run "make submit" to create a file called "hw03_<*userid*>.tar.gz", where <*userid*> is your gmail userid. Upload this file on myCourses under Content, Homework Submissions, Homework 02 Submission. This assignment is due at 11:59 PM on Sunday, February 12, 2017. You may submit as many times as you wish; but only the latest submission will be graded.

This assignment is worth 10 points. Your grade will be calculated as follows:

- If you submit late without an extension, there will be a two-point deduction for every 24 hours you are late. Extensions are available only in special circumstances, and can only be given by the instructor or a TA.
- There will be a three-point deduction for submissions that do not follow the required format. For instance, if you do not run make submit to create the correct tar file, or if you run make submit on a machine where your userid is incorrect so that the resulting tar file has the wrong name OR wrong contents (your userid is used to create a sub directory that is contained in the tar file).
- There will be a six-point deduction if there are compiler errors when compiling your code.
- There will be a three-point deduction if your code does not run to completion on any of the test cases used to grade your code. For instance, if your code causes a segmentation violation, there will be a three-point deduction.
- There will be a two-point deduction for each class of compiler warning message issued when compiling your code.
- There will be a two-point deduction if matMult does not print the correct results for the test cases in the Makefile test, and a one-point deduction for each of four unpublished test cases on matMult.