CSCI 2500 — Computer Organization
Homework 1 (document version 1.3)
Matrix Multiplication in C

Overview

- This homework is due by 11:59:59 PM on Friday, September 20, 2019.
- This homework is to be completed **individually**. Do not share your code with anyone else.
- You **must** use C for this homework assignment, and your code **must** successfully compile via `gcc` with absolutely no warning messages when the `-Wall` (i.e., warn all) compiler option is used. We will also use `-Werror`, which will treat all warnings as critical errors.

Homework Specifications

In this first individual homework assignment, you will implement matrix multiplication. (For you Math dual majors, you're welcome.) If you need a refresher in how matrix multiplication works, look in a math textbook or check out Wikipedia. Note that we will revisit this topic in the future, but for now you will implement your algorithm in C.

Command-Line Arguments

Four command-line arguments are required (i.e., `argv[1]` through `argv[4]`). The first two command-line arguments specify how many rows and columns are in the first matrix, and the second two command-line arguments specify how many rows and columns are in the second matrix. To properly translate the command-line arguments, use the `atoi()` function, which you can learn more about by checking out the `man` page.

Once the pair of matrix dimensions are identified and validated, prompt the user to enter the values (non-negative integers) of each matrix. Use `scanf()` to read in the appropriate values.

Example Program Execution

On the next page are example program executions that you can use to better understand how your program should work and how you can begin to test your code. Note that your program must properly return either `EXIT_SUCCESS` or `EXIT_FAILURE`, which you can verify via `echo $?`.

Also, be sure to match the output formatting exactly as shown (to ensure full credit on Submitty). In particular, when displaying a matrix, each line must start with '`[`' and end with '`]`' (as shown below). Further, right justify and vertically line up your columns as follows:

```
[12 34 5567]
[ 8  9  123]
[45 67    8]
[ 9 10   11]
```

Example program executions are shown below:

```
bash$ ./a.out 2 3 3 1
Please enter the values for the first matrix (2x3):
1 2 3
4 5 6
Please enter the values for the second matrix (3x1):
7
8
9

[1 2 3]
[4 5 6]
multiplied by
[7]
[8]
[9]
equals
[ 50]
[122]
bash$ echo $?
0
bash$ ./a.out 4 5 6 7
ERROR: Invalid inputs!
bash$ echo $?
1
bash$ ./a.out 2 2 2 2
Please enter the values for the first matrix (2x2):
1 2
3 4
Please enter the values for the second matrix (2x2):
5 6
7 8

[1 2]
[3 4]
multiplied by
[5 6]
[7 8]
equals
[19 22]
[43 50]
bash$ echo $?
0
```

Starting Point

Below is a starting point for your code (e.g., `hw1.c`):

```
/* hw1.c */
/* NAME: <your-name-here> */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main( int argc, char * argv[] )
{
  /* Ensure we have the correct number of command-line arguments */
  if ( argc != 5 )
  {
    fprintf( stderr, "ERROR: Invalid inputs!\n" );
    return EXIT_FAILURE;
  }



  return EXIT_SUCCESS;
}
```

Submission Instructions

Before you submit your code, be sure that you have clearly commented your code (this should not be an after-thought). Further, your code should have a clear and logical organization. In general, each function should easily fit on a single screen and have a clear (and clearly documented) purpose. Variable names and function names should be intuitive and meaningful. And use a consistent logical method for indentation.

To submit your assignment (and also perform final testing of your code), please use Submitty.

Note that the test cases for this assignment will be available on Submitty a few days before the due date and will include hidden test cases.

Also as a reminder, your code **must** successfully compile and run on Submitty, which uses Ubuntu v18.04.3 LTS.