

Programming Assignment #5*

Due date: 2/21/17 11:59pm

Programs are to be submitted to Gradescope by the due date. You may work alone or in groups of two. Programs submitted up to 24 hours late will still be accepted but incur a 10% grade penalty. Uploading your programs to gradescope will immediately score your submission.

Your program grade will be the score of the *last* submission that you have uploaded.

Programs must compile using `gcc -Wall` without any warnings. Each program that compiles with a warning will incur a 10% grade penalty. **Each program will have 5 seconds to compile and run test cases on gradescope.**

In this assignment, you will download `Program5Source.zip` and complete the code fragments as described below.

Problem 1: `deletespaces.c` (50 points, 5 per test case)

In this problem you will write a program that reads a string, deletes any whitespace from the string, and prints the result back to the user. Read the string using `fgets`, a safe way to read in strings. To read a string of length at most `n` into `char* str`, use the following:

```
fgets(str, n + 1, stdin);
```

Generally, `fgets` is used to read a string from a file, but using `stdin` as the third argument causes `fgets` to read a string from `stdin` instead.

Example output:

```
[rsgysel@pc17 ~]\$ ./deletespaces
Enter a max string length: 99
Enter the string to delete whitespace from: The Quick Brown Fox Jumps Over The Lazy Dog
Result:
TheQuickBrownFoxJumpsOverTheLazyDog

[rsgysel@pc17 ~]\$ ./deletespaces
Enter a max string length: 20
Enter the string to delete whitespace from: a bcd ef g hi
Result:
abcdefghijkl
```

*Last updated February 15, 2017

Problem 2: matrixaddition.c (50 points, 5 per test case)

In this problem you will implement integer matrix addition. This will require you to dynamically allocate space for a 2 dimensional array. An example of matrix addition is:

$$A + B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 8 & 5 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1+1 & 2+0 & 3+1 \\ 0+0 & 8+1 & 5+0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 \\ 0 & 9 & 5 \end{bmatrix} = C$$

In general, if A and B are matrices with the same number of rows and columns, then their sum $C = A + B$ is defined by $C[i][j] = A[i][j] + B[i][j]$. Here, $A[i][j]$ is the integer in the i^{th} row and j^{th} column of A . In the example above, $A[1][1] = 1$, $A[1][3] = 3$, and $A[2][2] = 8$.

This program has been partially written for you in `matrixaddition.c`. Write the body of functions that are marked with a comment that begins with

```
// Homework TODO: ...
```

Do not modify other parts of the code.

Example output:

```
[rsgysel@pc17 ~]\$ ./matrixaddition
Enter the number of rows and columns: 2 2
Enter matrix A:
Input row 0 elements, separated by spaces: 1 0
Input row 1 elements, separated by spaces: 0 1
Enter matrix B:
Input row 0 elements, separated by spaces: 2 2
Input row 1 elements, separated by spaces: 3 4
A + B =
3 2
3 5

[rsgysel@pc17 ~]\$ ./matrixaddition
Enter the number of rows and columns: 3 6
Enter matrix A:
Input row 0 elements, separated by spaces: 1 2 3 4 5 6
Input row 1 elements, separated by spaces: 0 0 0 0 0 0
Input row 2 elements, separated by spaces: 10 5 23 84 91 2
Enter matrix B:
Input row 0 elements, separated by spaces: 9 8 7 6 5 4
Input row 1 elements, separated by spaces: 2 1 2 1 2 1
Input row 2 elements, separated by spaces: 14 83 62 93 45 32
A + B =
10 10 10 10 10 10
2 1 2 1 2 1
24 88 85 177 136 34
```