CSE 240 Spring 2021 Homework 3, 2D Arrays (50 points)

Due Saturday, February 6, 2021 at 11:59PM, plus a 24-Hour grace period

Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures, including basic C syntax, using the strings, arrays, and multidimensional arrays. By the end of the assignment, you should have

- understood the concepts of data, data declaration, forward declaration, and manipulation of values stored in the memory.
- reviewed the array operations that you have learned in your previous programming classes;
- written a program using multidimensional arrays in C.

Note: **Do not** use pointer in this assignment. Pointer assignment will be given in the next homework.

Reading: Textbook Chapter 2, sections 2.1, 2.2, 2.3, and 2.4, and lecture slides covered.

Exercising: Complete the multiple choice questions in Textbook Section 2.12. The answers of the questions are available in course Web page.

You are expected to do the majority of the assignment outside the class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. However, do not share your answers or code in the course discussion board. **Do not cooperate with your peers in doing the individual assignments.**

Programming Assignment

For this assignment, you can use ASU General GNU GCC environment or Visual Studio. Please specify the environment that you use at the beginning of your program:
```
// State the IDE that you use: Visual Studio or GCC
```
If you use any other environment, please test it either using GCC or visual Studio before submission. If you do not specify the environment, the grader will use either GCC or Visual Studio, but not both. If you program does not compile or execute correctly, your grade will be deducted. **GCC command to compile this program: *gcc -std=c99 -o output hw03q1.c***
Execute with: *./output*

1. You are given hw03q1.c file, which contains a partially completed program. You are to follow the instructions contained in comments and complete the required functions. You will be writing functions for a program that does manipulation on 1-D and 2-D arrays. In one part, the program defines several common operations on 2D integer arrays (AKA matrices). In another function, you are asked to reverse one long user-input string and then count the number of occurrences of a character in that string. Example output given below. Go through the C file and see the sample expected output below.   [50 points]

CSE240 HW3: 2D Integer Arrays

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

2 4 6 8 10
12 14 16 18 20
22 24 26 28 30
32 34 36 38 40
42 44 46 48 50

1 49 169 361 625

2 8 18 32 50
72 98 128 162 200
242 288 338 392 450
512 578 648 722 800
882 968 1058 1152 1250

1 36 121 256 441
4 49 144 289 484
9 64 169 324 529
16 81 196 361 576
25 100 225 400 625

1 4 9 16 25
36 49 64 81 100
121 144 169 196 225
256 289 324 361 400
441 484 529 576 625

```
CSE240 HW3: 2D Character Arrays


Enter sentence with max 5 words: Did Tom eat the rat

Reveresed string:
tar eht tae moT diD
No of occurences of a: 2
```

Note, when you create the Visual Studio project, the default location of project is similar to the file path shown in example figure above. For your convenience, you may change the project location while creating the project at the beginning in New Project dialog box.

Grading and Rubrics

Each sub-question (programming tasks) has been assigned certain points. We will grade your programs following these steps:
 (1) Compile the code. If it does not compile, 50% of the points given for the code under compilation will be deducted. Then, we will read the code and give points between 50% and 0, as shown in right part of the rubric table.
(2) If the code passes the compilation, we will execute and test the code using test cases. We will assign points based on the left part of the rubric table.
In both cases (passing compilation and failed compilation), we will read your program and give points based on the points allocated to each sub-question, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.
Please notice that we will not debug your program to figure out how big or how small the error is. You may lose 50% of your points for a small error such missing a comma or a space!
We will apply the following rubrics to **each sub-question** listed in the assignment. Assume that points assigned to a sub-question is *pts*:

Rubric Table

Major	Code passed compilation				Code failed compilation		
Points	pts * 100%	pts * 90%	pts * 80%	pts * 70% - 60%	pts * 50% - 40%	pts * 30% - 10%	0
Each sub-question	Meeting all requirements, well commented, and working	Working correctly in all test cases. Comments not	Working with minor problem, such as not writing comments,	Working in most test cases, but with major	Failed compilation or not working correctly, but showing	Failed compilation, showing some effort, but the code does not	No attempt

	correctly in all test cases	provided to explain what each part of code does.	code not working in certain uncommon boundary conditions.	problem, such as the code fail a common test case	serious effort in addressing the problem.	implement the required work.	

What to Submit?

This homework assignment will have multiple parts. You are required to submit your solutions in a compressed format (.zip).   Make sure your compressed file is label correctly - lastname_firstname**3**.zip.   (All lowercase, do not put anything else in the name like "hw2".)

The compressed file MUST contain the following:

hw03q1.c

No other files should be in the compressed folder.
If multiple submissions are made, the most recent submission will be graded. (Even if the assignment is submitted late.)

Submission preparation notice: The assignment consists of multiple files. You must copy these files into a single folder for canvas submission. To make sure that you have all the files included in the zip file and they work after unzip operation, you must test them before submission. You must also download your own submission from the canvas. Unzip the file on a different machine, and test your assignment and see if you can open and test the files in a different location, because the TA will test your application on a different machine. If you submitted an empty project folder, an incomplete project folder, or a wrong folder, you cannot resubmit after the submission linked is closed! We grade only what you submitted in the canvas. We cannot grade the assignment on your computer or any other storage, even if the modification date indicated that the files were created before the submission due dates. The canvas submission may take a few minutes. Be patient and wait for it to complete.

Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

Late submission deduction policy

- No penalty for late submissions that are received within 24 hours after the deadline (before Sunday midnight);

- 10% grade deduction for every day it is late after the grace period (After Sunday);
- No late submission after Tuesday at 11:59PM.

Academic Integrity and Honor Code.

You are encouraged to cooperate in study group on learning the course materials. However, you may not cooperate on preparing the individual assignments. Anything that you turn in must be your own work: You must write up your own solution with your own understanding. If you use an idea that is found in a book or from other sources, or that was developed by someone else or jointly with some group, make sure you acknowledge the source and/or the names of the persons in the write-up for each problem. When you help your peers, you should never show your work to them. All assignment questions must be asked in the course discussion board. Asking assignment questions or making your assignment available in public websites before the assignment is due will be considered cheating.