CSC 111 - FALL 2023
FUNDAMENTALS OF PROGRAMMING
WITH ENGINEERING APPLICATIONS
ASSIGNMENT 4
UNIVERSITY OF VICTORIA

Due: Sunday, Oct. 29th, 2023 at 11:59pm. **Late assignments will not be accepted.**

This assignment will be submitted electronically through BrightSpace (as described in 'Submission Instructions' below). All code submissions must be your own work. Sharing code with other students (as either sender or receiver) is plagiarism. Use of generative AI tools (like ChatGPT) for any aspect of programming assignments is also prohibited.

1 Overview

This assignment covers arrays, libraries and applications of linear algebra to engineering programming. As with the previous assignment, this assignment is designed to emphasize the importance of rigorous testing and debugging.

A header file `matrix_and_vector.h`, containing function and datatype declarations for a linear algebra library, has been posted to BrightSpace, along with several tester programs that define a `main` function which calls the functions in the library. Your task is to implement a single `matrix_and_vector.c` source file containing definitions of the functions specified in `matrix_and_vector.h` (your code should contain the line '`#include "matrix_and_vector.h"`'). You can verify that your library implementation is complete and correct by compiling it in combination with one of the provided tester programs.

For example, to compile with the `vector_tester.c` program, you can use the command

```
gcc -Wall -std=c18 -o vector_tester vector_tester.c matrix_and_vector.c -lm
```

Note that you may not modify `matrix_and_vector.h` in any way. Your implementation must be completely contained in the separate `matrix_and_vector.c` file.

Your `matrix_and_vector.c` file **must not** contain a definition for `main()`, since your code is providing a library implementation, not a main program. If your submission contains a `main` function, it will receive a mark of zero.

Best Practices for Code Style

Your submissions **must** observe the following guidelines. Submissions which do not meet the guidelines will have marks deducted, even if they function correctly (although some of the guidelines, such as those relating to uninitialized memory, might also affect the ability of the program to work correctly on a reliable basis, which could result in a deduction for both style and function).

- Submitted files must be syntactically correct and named according to the assignment specification.
- Every source file must contain a comment (at or near the top of the file) with your name, student number, the date the code was written and a brief description of the program.
- The `goto` statement is not permitted in any assignment submissions. Instead, your code should use structured control flow elements, like `if`, `for` and `while`.
- Global variables (data variables created outside of the scope of a function) are not permitted, except when explicitly allowed by the assignment. For this assignment, no global variables are permitted, except for `const` variables if needed.
- Every function with a non-`void` return type must return a value.
- Uninitialized variables may not be used before being assigned a value.
- All variables, function parameters and function return types must be explicitly typed.
- Any dynamically allocated memory must be freed explicitly. For this assignment, it should not be necessary to use any dynamically allocated memory.

2 Function Specifications

The full specifications for most of the required functions are contained in the `matrix_and_vector.h` file (so make sure to read the comments in that file carefully before writing your solution). The sections below contain some extra detail for functions that require mathematical formulas (which are hard to typeset in a C comment).

In the context of linear algebra (not C programming), the individual elements of a vector $\vec{v}$ are numbered $v_1, v_2, \ldots, v_n$, but since this assignment will involve C arrays, the sections below will number the elements of vectors starting at 0 for symmetry with the expected implementation.

2.1 Dot Products

Consider two vectors $\vec{v} = (v_0, v_1, \ldots, v_{n-1})$ and $\vec{w} = (w_0, w_1, \ldots, w_{n-1})$. The **dot product** (or **inner product**) of $\vec{v}$ and $\vec{w}$, denoted by $\vec{v} \cdot \vec{w}$, is defined to be

$$\vec{v} \cdot \vec{w} = v_0 w_0 + v_1 w_1 + \ldots + v_{n-1} w_{n-1}.$$

2.2 Vector Norms

Consider a vector $\vec{v} = (v_0, v_1, \ldots, v_{n-1})$. The **norm**, or **length** of the vector $\vec{v}$, denoted $||\vec{v}||$, is defined to be

$$||\vec{v}|| = \sqrt{\vec{v} \cdot \vec{v}} = \sqrt{v_0^2 + v_1^2 + \ldots v_{n-1}^2}.$$

Recall that the `sqrt` function (from `math.h`) can be used to compute square roots in C programs.

2.3 Angles Between Vectors

Consider two vectors $\vec{v} = (v_0, v_1, \ldots, v_{n-1})$ and $\vec{w} = (w_0, w_1, \ldots, w_{n-1})$, and assume that both vectors have a non-zero norm (i.e. $||\vec{v}|| > 0$ and $||\vec{w}|| > 0$).

An identity from linear algebra (which we will use, but otherwise will not be examinable in this course) gives

$$\vec{v} \cdot \vec{w} = ||\vec{v}|| \cdot ||\vec{w}|| \cdot \cos\theta,$$

where θ is the angle between the vectors $\vec{v}$ and $\vec{w}$. Solving this equation for θ gives the following expression to compute the angle between two vectors.

$$\theta = \arccos\left(\frac{\vec{v} \cdot \vec{w}}{||\vec{v}|| \cdot ||\vec{w}||}\right).$$

The inverse cosine function (arccos) is available in `math.h` as the `acos` function.

2.4 Outer Products

Consider two vectors $\vec{v} = (v_0, v_1, \ldots, v_{n-1})$ and $\vec{w} = (w_0, w_1, \ldots, w_{m-1})$. The **outer product** of $\vec{v}$ and $\vec{w}$, denoted by $\vec{v} \otimes \vec{w}$, is an $n \times m$ matrix defined to be

$$\vec{v} \otimes \vec{w} = \begin{pmatrix} v_0 w_0 & v_0 w_1 & \cdots & v_0 w_{m-1} \\ v_1 w_0 & v_1 w_1 & \cdots & v_1 w_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n-1} w_0 & v_{n-1} w_1 & \cdots & v_{n-1} w_{m-1} \end{pmatrix}$$

3 Evaluation

Your code must compile and run correctly in the CSC 111 virtual lab environment using the compile and run commands given above. If your code does not compile **as submitted** (with no modifications whatsoever) when combined with the provided `matrix_and_vector.h` file and a suitable main program, you will receive a mark of zero. Note that you are not permitted to change anything in the `matrix_and_vector.h` file, since only the `matrix_and_vector.c` file will be collected for marking, so you must make sure that your function definitions comply with the specification in the provided version of the `matrix_and_vector.h` file.

This assignment is worth 4% of your final grade and will be marked out of 21, with one mark for code style (as usual) and the remaining marks distributed among the different library functions as follows.

- The `set_vector`, `mul_vector_by_scalar`, `dot_product`, `norm`, `angle_between_vectors`, `add_vectors`, `identity`, `matrices_equal`, `add_matrices`, `trace`, `transpose`, `omit_row`, `omit_column`, `outer_product` and `circulant` functions are each worth 1 mark.
- The `matrix_vector_multiply` function is worth 2 marks.
- The `matrix_multiply` function is worth 3 marks.

Second Chance Submission

After the initial due date (Sunday, Oct. 29th, 2023) has passed, your code will be automatically tested using a variety of inputs (which will not necessarily match the example inputs shown in the previous sections). If your programs do not give correct output on some of the tested inputs, you will be allowed to fix your program and resubmit it until Wednesday, Nov. 1st, 2023. The grade for the assignment will be computed to be the average of the grade for both submissions (so you can recover some, but not all, of your marks by fixing and resubmitting your code). If you do not resubmit your code, your initial submission will be used in place of the resubmission.

If you do not submit an initial version of your code before the due date, you are **not** permitted to resubmit your code and will receive a mark of zero.

Submission Instructions

All submissions for this assignment will be accepted electronically. Submit your `matrix_and_vector.c` file through the Assignment 4 entry on BrightSpace. You are permitted to delete and resubmit your assignment as many times as you want before the due date, but no submissions will be accepted after the due date has passed. For the second chance submission described above, a separate assignment entry will be created after the initial due date.

Ensure that each file you submit contains a comment with your name and student number, and that each file is named correctly (as described in this document). If you do not name your files correctly, or if you do not submit them electronically, it will not be possible to mark your submission and you will receive a mark of zero.

To verify that you submitted the correct file, you can download your submission from BrightSpace after submitting and test that it works correctly in the virtual lab environment. It will be assumed that all submissions have been tested this way, and therefore that there is no reasonable chance that an incorrect file was submitted accidentally, so no exceptions will be made to the due date as a result of apparently incorrect versions being submitted.