CIS7 Project Documentation Guide

In the documentation, provide at least 2 pages (single-space) that contains the following components of your course project:

* Team name, members.

Me, Myself, and I.

Joshua Hubbard

* Project Information and details: (30 points)
* What problems are you solving in this project?

Taking a 2d graph and turning the information into usable code. I am also facing the problem of how to use the slight variations of code to get exactly what i want.

* What solutions are you implementing in the project?

The solutions I am using is the code shown in labs 11.1-3. These include the addEdge fuction along with the Graph function, 2 very important pieces of code.

* Provide explanation of calculations and algorithm implementation.

The main thing I am trying to accomplish is to have the addEdge fuctions correctly display the goal of the selected option (Shortest path for example). Another important calculation im trying to impliment is the matrix that contains not only all connections but also their weights.

* What is the program objectives? Explain how your program is interacting with the user and its purpose.

The program is a simple one in the fact that the user inputs a number 1-4 and will get a response back based on that number. Depending on the key pressed, the response will have something to do with a road map.

* How is discrete structures implemented in the C++ program?

The program heavily relies on discrete structures, with about 50% of the work being from chapter 11. Every option has its own void function dedicated to it, and all use the main graph function.

* What are the limitations of the program?

The main limitation i can think of is the fact that you can only make choices in an ascending order (1,2,4 not 1,4,2). This makes it so you have to rerun the program to make those selections again.

* Provide recommendation on improving the limitations of the program.

Adding someway to have any option you choose work no matter the order would be a great way to fix the issue.

* Flowchart AND Pseudocode. (30 points)
* Write the pseudocode for the program, from start to finish. Be sure to include decision-making branching.
* Use standard shapes for flowchart, be sure to include decision-making branching.