Programming in C++ comes with many challenges. One must be aware that the same implementations may not work everywhere. For this project, the project into 3 steps. First, classes were created for each of the files to be read. This enabled objects of the airport, airline and route files to be created. Next a file reading class was created. This class handled the reading of the datafiles for the project and updated hashmaps for easy retrieval later on in the program. Lastly there is a class for the route-finding part of the program where the search was implemented using the breadth-first search algorithm. This project also includes a Node class to represents nodes being used in search tree.

In total, 6 classes were made and a file for helper functions was created as well. The program is driven in the main.cpp file and has test cases included in the folder. The source and destination are retrieved from a user created input file and after a successful search is completed, the results are written to an output file. Once the destination airport is in the city and country of the user’s destination, the goal has been reached and this is captured in the isGoal method in the route-finding class.

For this project, I came to understand the importance of unit testing. As C++ is still new to me, it was important that every little change be tested before moving on to the next phase. Although I encountered many errors, this move number of errors by a significant amount. I also recognised the importance of writing a better user interface. An earlier approach to solving this problem had me making the user pass in the route, airline and airplane files as parameters. However, with this implementation, the user types the relative file path in the console. Once this is passed into the FindRoute object, the datafiles used in the project are read.

All in all, the project was a learning journey as it involved googling a lot of the different implementations of basic structures in C++ that I had not used before.

**References**

*C++ Deque*. (n.d.). https://www.programiz.com/cpp-programming/deque

*C++ Vectors (With Examples)*. (n.d.). https://www.programiz.com/cpp-programming/vectors

*Efficient way to delete a value from an unordered\_map, when given just the value C++*. (2015, May 19). Stack Overflow. https://stackoverflow.com/questions/30335156/efficient-way-to-delete-a-value-from-an-unordered-map-when-given-just-the-value

GeeksforGeeks. (2022, October 14). *unordered\_map in C++ STL*. https://www.geeksforgeeks.org/unordered\_map-in-cpp-stl/

*What’s the C++ version of Java’s ArrayList*. (2010, October 19). Stack Overflow. https://stackoverflow.com/questions/3971049/whats-the-c-version-of-javas-arraylist