|  |
| --- |
|  |

**Felix Prince-Agbodjan**

**35382024**

**Intermediate Computer Programming**

**Cohort A**

**Dr. Robert Sowah**

**November 29th, 2022**

Jordan and I first restarted our code from our first attempt to use object-oriented programming which turned out to be a huge bust with our IDE throwing many errors. With insight from counterparts, we attempted implementing lists to store information about airports and their Id’s and their respective connections only to later on realize that this process would be much complicated in C++ and we would be giving ourselves the unnecessary extra work for no reason.

Our hallelujah solution was to implement vectors in an unordered\_map. Where these vectors would be storing the start city and destination city and their airportids, and the countries that have more than one airport into different vectors and call them within preceding functions.

We also created a graph to store information about the routes and airports and use the breath-first search approach to travers through the graph to find direct and indirect flights. Direct flights were determined by storing the start country, start city, destination, country and destination city in a 2 by 2 vector and got the airportids of the start city and start country into a vector called start\_airportiDs of type string, the exact process was performed on the destination country and destination city into a vector called des\_airportIDs of type string.

All the airports cities and countries were to be referenced by the airportIDs within the unordered\_map which is why we are working mostly with the ids so that when the bfs traverses through the graph to retrieve those airportids, it references those airport names from the unordered\_map and returns them as well.

Indirect flight were determined by those airportids that did were not present in the vectors of the start\_airportiDs. We also parsed our input file in our main function and after the search based on the input file was computed the required information being the airport country and the flights and total number of flights accompanied with the number of stops was written to our output file.

In Conclusion, working with different data structures was quite the insightful ride. C++ is quite the tedious language to work with but I must admit it wasn’t fun to handle because of how irritable the errors get.