**ICP Individual Project 1**

This project aimed to find a flight or list of flights, given a start country and city and a destination country and city. The user input was to be given by a file and the output was to be printed into a file.

I created a Route class that contains constructors for a route object. A route object contains the source and destination airport IATA codes, their ID’s, and the stops for each route. The Route class has a method, getRoutes, that reads the routes file into a Hashmap called routeMap. Its’ key is the IATA of a given source airport. The value for each key is an Arraylist that has the IATA codes of all destination airports that can be reached from the key. Inside the route class is a method, findRoute, that implements a breadth-first search on the elements in routeMap by taking the IATA codes of the source and destination airports as arguments.

The solution path returned by findRoute is defined in a Node class. This class has a constructor that creates node objects to handle the parent-child relationship between each airport. The constructor takes the IATA code of the current airport and the node of its parent as parameters. The solution path method within this class returns an array containing the IATA codes of each airport on the path from the goal to the destination. The names of the airports are obtained from the objectInit method in the Airport class.

The Airport class contains a constructor that creates airport objects for each airport in the data file. The airport objects are stored as values in a hashmap and the key of each value is an array list that contains the IATA code, city, and country of the object it indexes. The objectInit method in this class allows for the initialized airport object to be accessed using the string value of the IATA codes. It returns an airport object when its IATA is passed as an argument.

The readWrite class contains methods for reading the input file and writing to the output file. The fileReader method reads the lines of an input file into an array list. It then passes the array list to the inputConv method within the class. The inputConv method splits the strings into separate start and destination city and country arguments. It then uses the arguments to access the corresponding start and destination airports within the airportsMap hashmap and passes the IATA for the airports as corresponding start and end arguments to the findRoutes method in the route class. When a solution is found, the solution path array list is passed as an argument to the fileWriter method which writes the final flight path, including the number of stops, into a new file .