**Approach:**

- I first created instance variables to read our databases and input files and detect possible arrival and departure points. And see the destination path, and determine the validity of our start and end points (i.e., verify if there is a viable path one can take to reach the destination) which is assumed to be accurate. And file objects for input and output files and array list to read routes.csv data.
- I proceeded to read the database files as well as create our output and input files through the constructor.
- I then created the set_input_data method that writes start and information on our input file regardless if it exists or not as specified by the user and calls our set_start_goal method.
- The set_start_goal method will read the start and end information of our input file, call the viewStart_information to display information in our input file, and determine all possible arrival and departure airports and validity of our destination path.
- We then read information on our routes database and assign it to our route list via the setRoute method.
- Our find_goal_path method will then copy all our possible departure points to the airport list, call our setRoute method, and traverse through our airport list if it's not empty. And a possible valid destination path exists, or our goal destination hasn't been found.

  The goal destination is traversing through the airport list and locating various arrival points from the starting point. And determine if our airport is our destination point. And, if not, our destination point, add our arrival airports to our airport list and path dictionary that stores needed data of our arrival point and repeat the process.
- Our getRoute_information method will then detect the path information taken from destination point to start point stored on our path dictionary and store the information on our routes_information list and reverse it because we started from our ending point.
- Our get_Output_data and getOutput_info will then respectively write a valid route path on our output file or write the prompt for the case of a non-existent valid route path.

**Lessons learnt:**

- You can’t use enhanced for to traverse through a list and update it at the same time as it will lead to errors.
- Dividing your program sections into various functions or methods is always the best way to debug.
- boolean_expression == true and boolean_expression == false can easily be edited to easily be boolean_expression and !boolean_expression respectively.
- How to create and enter data into a dictionary in java.

**Citation list:**

Java HashMap. (n.d.). Retrieved September 30, 2022, from https://www.w3schools.com/java/java_hashmap.asp