Homework 9

Released Friday

Due Friday Oct 28th , 11:59pm

Goals:

- Structs:
    - Initializing and accessing structs appropriately

Task: Managing an airport

In this assignment, you will assist in managing an airline company by creating a piece of software that allows the company to:

- add an airport
- add destinations to an airport
- check if there is a one stop flight from one airport to another
- check if there is a two stop flight from one airport to another

Given:

- Airport_Manager.h - header file containing all methods that need to be implemented

You will not be given a .c file, a main file, or a Makefile, so make sure to write tests to check to make sure your program functions correctly.

Make sure that your .c file is called "**Airport_Manager.c**"

`void createAirportArray()` initializes all global variables. The maximum number of airports should be initially set to 20.

`int hasOneStopFlight(Airport * start, Airport *dest)` returns 1 if there is a one stop flight from start to dest, 0 if there is not, or -1 if the arguments are invalid.

`int hasTwoStopFlight(Airport * start, Airport *dest)` returns 2 if there is a one stop flight from start to dest, 1 if there is not a one stop flight but there is a two stop flight, 0, if there is not a one stop or a two stop flight from start to dest, or -1 if the arguments are invalid. For example, if there is no direct flight from Chicago to New York, but there is a direct flight from Chicago to Atlanta and a direct flight from Atlanta to New York, then there is a two stop flight from Chicago to New York.

`int addDestination(Airport * airport, Airport * dest)` returns 1 if a destination is successfully added to an airport's destinations array, 0 if the airport already flies to the specified destination, or -1 if the arguments are invalid.

`int addAirport(Airport * airport)` returns 1 if an airport is successfully added to the first empty entry in the airports array, 0 if the airports array already contains the specified airport, or -1 if the arguments are invalid. This function resizes the array by double if the number of airports is equal to the max number of airports.

`Airport * createAirport(const char * name)` returns a new airport with the name passed in as an argument. The function must store a copy of the name passed as an argument in the new airport. It is assumed that there will be no two airports with the same name.

`void printAirports()` prints all of the airports in the airports array in the format
"NumberOfAirports: <number of airports"
"<Airport Name>"
"<Airport Name>"
"<newline>"

`void freeAllAirports()` frees all memory allocated by the airport manager

Vocaruem Submission:

- **Airport_Manager.c**