Project- 2

The project is to load data through a CSV file and write a program that will conduct a variety of queries on that data.

Dataset

The dataset (attached) is a CSV file on all International flight departing from US Airports between January and June 2024 reported by the US Department of Transportation (https://data.transportation.gov/Aviation/International_Report_Passengers/xgub-n9bw). Each record holds a route (origin to destination) operated by an airline. This CSV file was modified to keep it simple and relatively smaller. Here is a description of each column:

- Column 1 – Month (1 – January, 2 – February, 3 – March, 4 – April, 5 – May, 6 – June)
- Column 2 – 3-letter IATA Airport Code for the origin airport (e.g., SAT for San Antonio International Airport)
- Column 3 – 3-letter IATA Airport Code for the destination airport
- Column 4 – 2/3-letter IATA Airline Code for the airline (e.g., AA for American Airlines).
- Column 5 – Total number of passengers in that month for that route

Note that there is a header row you must skip. Since this data holds passenger statistics for each route operated by an airline for 6 months, you should see the airline route repeated at most 6 times. For example, you will see the JFK to LHR operated by BA route 6 times, once for each of the 6 months.

Task 1 – create `route-records.h`

All data is loaded into an array of **`RouteRecord`**'s which will be queried in **`main()`**.

- Create a **`struct`** named **`RouteRecord`** that will hold information about a route that is operated by one airline. The struct will have the following data members:
    - Origin airport code
    - Destination airport code
    - Airline code
    - Array of passenger counts. There is 6 months' worth of data for each route. (Index 0 will represent January's passenger count, Index 1 will represent February's passenger count, etc.).
- Add the header guards and prototypes for the functions (see Task 2)

- Include this **enum** in your header file so you can use as values for determining what type of search you will conduct.
  ```
  typedef enum SearchType { ROUTE, ORIGIN, DESTINATION, AIRLINE } SearchType;
  ```

Task 2 – create route-records.c

Write the following functions:

- **RouteRecord* createRecords(FILE* fileIn)** – This function creates the array of **RouteRecord**'s and initializes it. The function takes in a file pointer. The function will do the following:
  - This function goes through the CSV file and counts the number of total records (not including the header)
  - Dynamically allocate memory for an array of **RouteRecord**'s based on the count.
  - Each **RouteRecord** struct object has an array of 6 integers to hold the number of passengers for 6 months. Initialize each of these integer values to 0. You do not need to initialize the other data members in the struct.
  - **Rewind** the file pointer.
  - Return the pointer to the array you dynamically allocated.
- **int fillRecords(RouteRecord* r, FILE* fileIn)** – This function will process the data in the CSV file. Essentially, the code will go through each record, parse out the record, and enter it into the array. The function will follow these rules:
  - The function will first see the IATA Airline Code and see its length.
    - If the length of the code is 3 (e.g., 04Q), the record is to be skipped.
    - If the length of the code is 2 (e.g., AA), continue to the next step.
  - The function will call **findAirlineRoute()** to see if the exact route with the origin, destination, and airline was already entered in the array. If it was found, then you will update the existing record in your array with the passenger data for that month. Recall there should be 6 entries (one for each month) for each route operated by an airline. If the route operated by the airline does not already exist in the array, add this new route to the array.
  - The function returns the actual number of **RouteRecord**'s used in the array. The value returned will be less than the size of the array created since not all records in the original CSV file will be entered into the array.
- **int findAirlineRoute(RouteRecord* r, int length, const char* origin, const char* destination, const char* airline, int curIdx)** – This **RECURSIVE** function finds a record in the **RouteRecord** array with the same origin and destination airport codes and airline. It returns the index number in which these three strings appear in the array. The function will return -1 if it cannot find these three strings in the same **struct** object.
- **void searchRecords(RouteRecord* r, int length, const char* key1, const char* key2, SearchType st)** – This function searches the **RouteRecord** array and prints out the results of the search.

    - You will traverse the array and compare specific data members against the keys.
    - The parameter st determines if the function is searching by **`ROUTE, ORIGIN, DESTINATION, AIRLINE`**.
    - For **`ORIGIN, DESTINATION, AIRLINE, key1`** will hold the value you are looking for. For **`ROUTE`**, you are searching both the origin and destination and airport, so **`key1`** and **`key2`** will hold those values, respectively, that you will use to compare against the data members. For example, if the search is by the destination: **`st`** will be equal to **`DESTINATION`**, **`key1`** will have an airport code that the user entered, and you will compare each struct's destination data member against the airport code.
    - You will print out the airline and the route for each matching value. Then, you will print out the total number of passengers on all matching records, total number of passengers by month for all matching records, as well as average numbers of passengers per month. Note that you must handle any instances where the search has 0 results.
- **`void printMenu()`** – This function prints the menu. Here is the function below. Be sure to add this prototype to the header file.

```
void printMenu()
{
     printf( "\n\n######### Airline Route Records
     Database MENU #########\n" );
     printf( "1. Search by Route\n" );
     printf( "2. Search by Origin Airport\n" );
     printf( "3. Search by Destination Airport\n" );
     printf( "4. Search by Airline\n" );
     printf( "5. Quit\n" );
     printf( "Enter your selection: " );
}
```

TASK 3: Complete the `main.c`

- Download the attached **`main.c`**
- Follow the instructions written in the comments in the **`main()`** function.
- The **`main()`** is the driver of the program. It calls the functions above to load the data from the CSV file and to run queries that the user asks for.
- The name of the file will be passed in through **command line arguments**.
- The user will enter a numeric value from the menu. You must handle the case in which the user enters invalid values (e.g., strings).

Task 4: Create a `makefile`

Create a **`makefile`** to compile and link all the files together. The grader will compile your code using your **`makefile`**.

Submission

Be sure that your code follows the class coding style requirements. Your output should be similar in format as compared to the sample output shown below. Create a folder named your abc123, place all program files in this folder. Zip the folder and submit this abc123.zip file.

Rubric

General Requirements – submission must have all files as per requirement and the code must compile properly without any errors. If not, the submission will be returned with a score of 0.

[05 points] - makefile
- [5 points] - make compiles and creates the executable

[10 points] - main.c
- [2 points] - command-line argument for input file name
- [2 points] - Open files and checks for success in opening
- [3 points] - Processes user input, handling bad input
- [3 points] - asks user for keys and calls appropriate functions for desired functionality

[05 points] - route-records.h
- [1 points] - header guard
- [2 points] - struct
- [2 points] - function prototypes

[30 points] - route-records.c
- [5 points] - createRecords()
- [5 points] - fillRecords()
- [8 points] - findAirlineRoute()
- [12 points] - searchRecords()

SAMPLE OUTPUT

```
Opening data-2024.csv...
Unique routes operated by airlines: 3541


######### Airline Route Records Database MENU #########
1. Search by Route
2. Search by Origin Airport
3. Search by Destination Airport
4. Search by Airline
5. Quit
Enter your selection: 1
Enter origin: LAX
Enter destination: LHR

Searching by route...
BA (LAX-LHR) VS (LAX-LHR) DL (LAX-LHR) AA (LAX-LHR) UA (LAX-LHR)
```

5 matches were found.

Statistics
Total Passengers: 770464
Total Passengers in Month 1: 110078
Total Passengers in Month 2: 85316
Total Passengers in Month 3: 133704
Total Passengers in Month 4: 133671
Total Passengers in Month 5: 145611
Total Passengers in Month 6: 162084

Average Passengers per Month: 128410

######### Airline Route Records Database MENU #########
1. Search by Route
2. Search by Origin Airport
3. Search by Destination Airport
4. Search by Airline
5. Quit
Enter your selection: 2
Enter origin: SAT
Search by origin...
VB (SAT-BJX) UA (SAT-BZE) WN (SAT-CUN) AS (SAT-CUN) UA (SAT-CUN) UA (SAT-GDL) Y4 (SAT-GDL) UA (SAT-LIR) 5D (SAT-MEX) VB (SAT-MEX) Y4 (SAT-MEX) VB (SAT-MTY) UA (SAT-NAS) UA (SAT-NRT) UA (SAT-RTB) NK (SAT-SAL) NK (SAT-XPL) UA (SAT-QRO) VB (SAT-QRO) WN (SAT-BZE) AA (SAT-CUN) AM (SAT-MEX) UA (SAT-MGA) AA (SAT-MLM) UA (SAT-PVR) UA (SAT-SJO) DL (SAT-YYZ) UA (SAT-GUA) AA (SAT-MEX) UA (SAT-SAL) UA (SAT-YVR) NK (SAT-CUN) SY (SAT-CUN) DE (SAT-FRA) AA (SAT-GDL) UA (SAT-LHR) UA (SAT-MEX) UA (SAT-MID) UA (SAT-PTY) AA (SAT-PVR) AA (SAT-SJD) AA (SAT-QRO) AA (SAT-BJX) AF (SAT-CDG) MQ (SAT-CUU) DL (SAT-MTY) AA (SAT-MTY) AA (SAT-SAL) YV (SAT-SLP) VB (SAT-TRC) 5D (SAT-NLU)
51 matches were found.

Statistics
Total Passengers: 334987
Total Passengers in Month 1: 55503
Total Passengers in Month 2: 42129
Total Passengers in Month 3: 59153
Total Passengers in Month 4: 49606
Total Passengers in Month 5: 58062
Total Passengers in Month 6: 70534

Average Passengers per Month: 55831

######### Airline Route Records Database MENU #########
1. Search by Route
2. Search by Origin Airport
3. Search by Destination Airport
4. Search by Airline
5. Quit
Enter your selection: 3
Enter destination: CUN
Searching by destination...
WN (ATL-CUN) DL (ATL-CUN) F9 (ATL-CUN) WN (AUS-CUN) AA (AUS-CUN) WN (BNA-CUN) AA (BNA-CUN) DL (BOS-CUN) AA (BOS-CUN) B6 (BOS-CUN) WN (BWI-CUN) NK (BWI-CUN) F9 (BWI-CUN) UA (CLE-CUN) F9 (CLE-CUN) AA (CLT-CUN) AA (CMH-CUN) DL (CVG-CUN) AA (CVG-CUN) F9 (CVG-CUN) VB (CVG-CUN) WN (DEN-CUN) UA (DEN-CUN) F9 (DEN-CUN) AA (DFW-CUN) NK (DFW-CUN) F9 (DFW-CUN) SY (DSM-CUN) DL (DTW-CUN) NK (DTW-CUN) F9 (DTW-CUN) UA (EWR-CUN) B6 (EWR-CUN) WN (FLL-CUN) B6 (FLL-CUN) NK (FLL-CUN) SY (GRI-CUN) WN (HOU-CUN) AA (IAD-CUN) UA (IAD-CUN) UA (IAH-CUN) NK (IAH-CUN) WN (IND-CUN) AA (IND-CUN) SY (IND-CUN) DL (JFK-CUN) AA (JFK-CUN) B6 (JFK-CUN) XP (LAN-CUN) AS (LAS-CUN) DL (LAX-CUN) AA (LAX-CUN) AS (LAX-CUN) UA (LAX-CUN) B6 (LAX-CUN) WN (MCI-CUN) AA (MCI-CUN) SY (MCI-CUN) F9 (MCI-CUN) WN (MCO-CUN) B6 (MCO-CUN) NK (MCO-CUN) F9 (MCO-CUN) RV (MCO-CUN) WN (MDW-CUN) F9 (MDW-CUN) AA (MIA-CUN) SY (MKE-CUN) DL (MSP-CUN) SY (MSP-CUN) F9 (MSP-CUN) WN (MSY-CUN) NK (MSY-CUN) SY (MSY-CUN) WN (ORD-CUN) AA (ORD-CUN) UA (ORD-CUN) NK (ORD-CUN) F9 (ORD-CUN) AS (PDX-CUN) AA (PHL-CUN) NK (PHL-CUN) F9 (PHL-CUN) WN (PHX-CUN) AA (PHX-CUN) WN (PIT-CUN) AA (PIT-CUN) G4 (PIT-CUN) DL (RDU-CUN) AA (RDU-CUN) B6 (RDU-CUN) AS (SAN-CUN) WN (SAT-CUN) AS (SAT-CUN) UA (SAT-CUN) DL (SEA-CUN) AS (SEA-CUN) AS (SFO-CUN) UA (SFO-CUN) DL (SLC-CUN) WN (STL-CUN) AA (STL-CUN) F9 (STL-CUN) WN (TPA-CUN) B6 (TPA-CUN) VB (BUF-CUN) VB (CMH-CUN) SY (FAR-CUN) SY (FSD-CUN) UA (GJT-CUN) UA (MCO-CUN) BA (MIA-CUN) DL (TPA-CUN) UA (AUS-CUN) WN (CMH-CUN) AA (IAH-CUN) VB (MCI-CUN) WN (MKE-CUN) DL (MSY-CUN) B6 (PBI-CUN) UA (PHX-CUN) AA (SAT-CUN) B6 (SAV-CUN) AA (ATL-CUN) UA (BWI-CUN) UA (CLT-CUN) G4 (CVG-CUN) F9 (IAH-CUN) F9 (MIA-CUN) UA (RIC-CUN) VB (BNA-CUN) AA (CHS-CUN) DL (CLE-CUN) VB (CLT-CUN) SY (DFW-CUN) AA (GRK-CUN) SY (HRL-CUN) SY (IAH-CUN) VB (MEM-CUN) AM (MIA-CUN) VB (MSY-CUN) AA (OKC-CUN) AA (RFD-CUN) NK (SAT-CUN) SY (SAT-CUN) F9 (TPA-CUN) UA (ABQ-CUN) DL (BWI-CUN) WN (COS-CUN) B6 (PHL-CUN) VB (PIT-CUN) WN (RSW-CUN) F9 (SJU-CUN)
153 matches were found.

Statistics
Total Passengers: 6899630
Total Passengers in Month 1: 1167104
Total Passengers in Month 2: 1156218
Total Passengers in Month 3: 1378843
Total Passengers in Month 4: 1046755
Total Passengers in Month 5: 1016041

Total Passengers in Month 6: 1134669

Average Passengers per Month: 1149938

######### Airline Route Records Database MENU #########
1. Search by Route
2. Search by Origin Airport
3. Search by Destination Airport
4. Search by Airline
5. Quit
Enter your selection: 4
Enter airline: AI
Search by airline...
AI (EWR-BOM) AI (EWR-DEL) AI (IAD-DEL) AI (JFK-BOM) AI (JFK-DEL) AI (ORD-DEL) AI (SFO-BLR) AI (SFO-BOM) AI (SFO-DEL)
9 matches were found.

Statistics
Total Passengers: 688717
Total Passengers in Month 1: 115763
Total Passengers in Month 2: 110867
Total Passengers in Month 3: 116901
Total Passengers in Month 4: 108172
Total Passengers in Month 5: 121597
Total Passengers in Month 6: 115417

Average Passengers per Month: 114786

######### Airline Route Records Database MENU #########
1. Search by Route
2. Search by Origin Airport
3. Search by Destination Airport
4. Search by Airline
5. Quit
Enter your selection: 5
Good-bye!