**Project - 2**



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

**Data Set**

The data set (attached) is a modified CSV file on all International flight departing from US Airports between January and June 2019 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-letter IATA Airline Code for the airline (e.g., AA for American Airlines). Some airlines will have a 3-letter airline code, your program will exclude them from the parsing.
* Column 5 – The passenger category, in our example, there is only one category.
* Column 6 – 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 six months, you should see the airline route repeated six times. For example, you will see the JFK to LHR operated by BA route 6 times, once for each of the six months.

**Task 1 – create route-records.h**

All data is loaded into an array of **RouteRecord**’s which will they 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 are six 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 } SearchTyp**

**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 six 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:
  + If the record contains an airline that has 3 letters, ignore this record and go to the next record.
  + 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 six 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 you the search has 0 results.
* **void printRecords( RouteRecord\* r, int length)** – This function prints the records. **r** denotes the pointer for the records and **length** is the number of records to be printed. You can also create another helper function that just prints one record - **void printRecord( RouteRecord r )**. The **printRecords** function can call the **printRecord** function to print one record at a time.
* **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 project2-main.c**

* Download the attached **project2-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 as well as attached in a separate txt file). Select all your program files and create a zip file out of that. Name this zip as **abc123.zip** file and submit on Blackboard.

**SAMPLE - OUTPUT**

Opening passenger-data.csv...

Unique routes operated by airlines: 3565

######### 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) AA (LAX-LHR) VS (LAX-LHR) NZ (LAX-LHR) UA (LAX-LHR)

5 matches were found.

Statistics

Total Passengers: 737697

Total Passengers in Month 1: 118092

Total Passengers in Month 2: 100836

Total Passengers in Month 3: 131221

Total Passengers in Month 4: 125954

Total Passengers in Month 5: 132702

Total Passengers in Month 6: 128892

Average Passengers per Month: 122949

######### 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: SFO

Search by origin...

AC (SFO-YYZ) CX (SFO-HKG) AF (SFO-CDG) BA (SFO-LHR) BR (SFO-TPE) UA (SFO-YVR) UA (SFO-FRA) UA (SFO-LHR) KL (SFO-AMS) PR (SFO-MNL) KE (SFO-ICN) EK (SFO-DXB) LH (SFO-FRA) UA (SFO-PVG) UA (SFO-SIN) VS (SFO-LHR) AM (SFO-MEX) CA (SFO-PEK) UA (SFO-ICN) UA (SFO-CUN) AC (SFO-YVR) CI (SFO-TPE) TK (SFO-IST) UA (SFO-TPE) UA (SFO-TLV) UA (SFO-HKG) AC (SFO-YUL) LX (SFO-ZRH) UA (SFO-NRT) EI (SFO-DUB) SQ (SFO-SIN) QF (SFO-SYD) AI (SFO-DEL) OZ (SFO-ICN) CM (SFO-PTY) MU (SFO-PVG) LH (SFO-MUC) WS (SFO-YVR) UA (SFO-CDG) UA (SFO-YYZ) TA (SFO-SAL) WS (SFO-YYC) UA (SFO-PEK) SQ (SFO-HKG) CZ (SFO-CAN) UA (SFO-MUC) SK (SFO-CPH) UA (SFO-MEX) UA (SFO-SYD) UA (SFO-SJD) NH (SFO-NRT) UA (SFO-YYC) DI (SFO-LGW) UA (SFO-AMS) NZ (SFO-AKL) UA (SFO-KIX) UA (SFO-PVR) UA (SFO-HND) AM (SFO-GDL) JL (SFO-HND) UA (SFO-ZRH) BF (SFO-ORY) AS (SFO-PVR) AY (SFO-HEL) HX (SFO-HKG) AS (SFO-SJD) TP (SFO-LIS) BF (SFO-PPT) FI (SFO-KEF) QK (SFO-YVR) QF (SFO-MEL) 4O (SFO-GDL) QK (SFO-YYC) MT (SFO-MAN) CZ (SFO-WUH) FJ (SFO-NAN) IB (SFO-MAD) UA (SFO-AKL) IG (SFO-MXP) LY (SFO-TLV) IB (SFO-BCN) UA (SFO-PPT) UA (SFO-CTU) 4O (SFO-CUN) QK (SFO-YEG) MU (SFO-TAO) SE (SFO-CDG) OO (SFO-YVR) OO (SFO-YYC) DL (SFO-YYZ) LH (SFO-STR) AF (SFO-KEF) UA (SFO-NGO) JL (SFO-NRT) AA (SFO-HKG) AM (SFO-TIJ) UA (SFO-STR) AM (SFO-HMO) 0Q (SFO-LBG) TK (SFO-ISL) AA (SFO-NRT) UA (SFO-BCN) KL (SFO-DUB) AM (SFO-PVR) RV (SFO-YVR) Y4 (SFO-GDL) Y4 (SFO-MEX) OO (SFO-YYJ) PR (SFO-CRK) CX (SFO-ICN) UA (SFO-MLM) AM (SFO-BJX) UA (SFO-BZE)

113 matches were found.

Statistics

Total Passengers: 7228389

Total Passengers in Month 1: 1137764

Total Passengers in Month 2: 966010

Total Passengers in Month 3: 1130858

Total Passengers in Month 4: 1192147

Total Passengers in Month 5: 1318673

Total Passengers in Month 6: 1482937

Average Passengers per Month: 1204731

######### 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: KIX

Searching by destination...

D7 (HNL-KIX) HA (HNL-KIX) JL (HNL-KIX) UA (SFO-KIX) JL (LAX-KIX) DL (HNL-KIX) DL (SEA-KIX) UA (GUM-KIX) 7C (GUM-KIX) CX (IAD-KIX) TR (HNL-KIX) CX (EWR-KIX)

12 matches were found.

Statistics

Total Passengers: 620418

Total Passengers in Month 1: 92839

Total Passengers in Month 2: 93977

Total Passengers in Month 3: 120276

Total Passengers in Month 4: 103034

Total Passengers in Month 5: 107544

Total Passengers in Month 6: 102748

Average Passengers per Month: 103403

######### 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: LO

Search by airline...

LO (ORD-WAW) LO (JFK-WAW) LO (LAX-WAW) LO (EWR-WAW) LO (MIA-WAW) LO (JFK-BUD) LO (ORD-KRK) LO (ORD-BUD) LO (EWR-RZE)

9 matches were found.

Statistics

Total Passengers: 327362

Total Passengers in Month 1: 34852

Total Passengers in Month 2: 29604

Total Passengers in Month 3: 45388

Total Passengers in Month 4: 57228

Total Passengers in Month 5: 67440

Total Passengers in Month 6: 92850

Average Passengers per Month: 54560

######### 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: 10

Invalid choice.

######### 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: asdf

Invalid input.

######### 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: asfd

Search by airline...

0 matches were found.

Statistics

Total Passengers: 0

Total Passengers in Month 1: 0

Total Passengers in Month 2: 0

Total Passengers in Month 3: 0

Total Passengers in Month 4: 0

Total Passengers in Month 5: 0

Total Passengers in Month 6: 0

Average Passengers per Month: 0

######### 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!