Lab #8

CS-2050 - Section B

Week of March 22, 2021

1 Requirements

In this lab, you will write a set of functions for maintaining a **sorted list** ADT. You will be provided with a library that contains a linked list implementation, which you may *optionally* use to implement your required function. Alternatively, you may use any list implementation of your choosing.

```
typedef struct {
        float distance;
        unsigned int flightNumber;
        unsigned short passengers;
} Flight;
```

1.1 compareFlights

```
int compareFlights(Flight *a, Flight *b);
```

Info: This function takes two pointers to **Flight** structs and compares them based upon their *flightNumber* members. The function should return a strictly negative number if a is *less than* b, 0 if they are equal, and a strictly positive number if a is *greater than* b.

1.2 insertFlightAscending

```
int insertFlightAscending(List *list, Flight *flight);
```

Info: This function takes a *list* and pointer to a struct. This function will insert the struct onto the list in *sorted order* using the above compare function. The function will maintain the list in **ascending order**.

1.3 printFlights

```
void printFlights(List *list);
```

Info: This function takes a list containing *Flight structs* and prints out all of the structs in a readable format.

1.4 countAllEqualFlights

```
int countAllEqualFlights(List *list, Flight *flight);
```

Info: This function takes a list containing struct pointers, and returns the number of structs on the list which are *equal to* the given struct.

2 Notice

Grading: Total 20 points

1. Write required *insert* function
   * 8 points
2. Write required *compare* function
   * 2 points
3. Write required *print* function
   * 4 points
4. Write required *count* function
   * 6 points

Notice:

1. All of your lab submissions must compile under GCC using the $-Wall$ and $-Werror$ flags to be considered for a grade.
2. You are expected to provide proper documentation in every lab submission, in the form of code comments. For an example of proper lab documentation and a clear description of our expectations, see the lab policy document.