```c
#include "header.h"

/**
 * Author: Tyler Tucker
 * Email: <tyler.m.tucker@okstate.edu>
 * Date: April 19th, 2020
 * Program Description: This file takes input from the server and then
    determines if a customer is in the summary file or not. The functions can
    also change information in the summary file and delete customers completely
 */


/*
    The addNewCustomer function takes a new customers information and apends
    it to the end of the summary.txt, it also sets the customer ticket number
*/
void addCustomer(struct clientInformation *customer, int new) {
    FILE* summary = fopen("Summary.txt", "a");

    if (new == 1) {
        customer->ticket = -1;
        customer->ticket = findCustomer(customer);
    }

    fprintf(summary, "%d, ", customer->ticket);
    fprintf(summary, "%s, ", customer->ClientName);
    fprintf(summary, "%s, ", customer->DateOfBirth);
    fprintf(summary, "%s, ", customer->Gender);
    fprintf(summary, "%d, ", customer->GovernmentID);
    fprintf(summary, "%s, ", customer->DateOfTravel);
    fprintf(summary, "%d, ", customer->NumberOfTravelers);
    fprintf(summary, "%s, ", customer->seats);

    if(new == 0){
        fprintf(summary, "%d, ", customer->server);
        fprintf(summary, "%s\n", customer->modified);
    }else{
        fprintf(summary, "%d\n", customer->server);
    }


    fclose(summary);
}


/*
    The changeOldCustomer function will take a returning customers information
    and changed what they have in the summary file to reflect the newer
    information
*/
void changeOldCustomer(struct clientInformation *customer){
    deleteCustomer(customer);
    addCustomer(customer,0);
}

/*
    The printCustomerInfo function will take a customer, find them in the
    summary file, and print their info to a buffer array
*/
void printCustomerInfo(struct clientInformation *customer, char *output){
    FILE* summary = fopen("Summary.txt", "r");
    char buffer[1024];
    int line = findCustomer(customer);
    int counter = 0;

    while((fgets(buffer, 1024, summary)) != NULL){
        if(counter == line){
```

```
68              strcpy(output, "Inquiry results: ");
69              strcat(output, buffer);
70              fclose(summary);
71              return;
72          }
73
74          counter++;
75      }
76      strcpy(output, "No results found");
77      fclose(summary);
78  }
79
80  /*
81      The deleteCustomer function will take a customer and look for it in the
82      summary file, it then deletes that person by writing everyone else to a temp
83      file and then deleting the summary file and renaming the temp file
84  */
85  void deleteCustomer(struct clientInformation *customer){
86      FILE* summary = fopen("Summary.txt", "r");
87      FILE* temp = fopen("temp.txt", "a");
88      char buffer[1024];
89      int counter = 0;
90      int line = findCustomer(customer);
91      printf("line: %d",line);
92      while((fgets(buffer, 1024, summary)) != NULL){
93          if(counter != line){
94              fprintf(temp, "%s", buffer);
95          }
96          counter++;
97      }
98
99      fclose(summary);
100     fclose(temp);
101     remove("summary.txt");
102     rename("temp.txt", "summary.txt");
103 }
104
105
106 /*
107     findCustomer looks at the summary file and tries to find a given ID
108     If that ID is found then it will return the line where the ID is
109     Otherwise the function returns -1
110 */
111 int findCustomer(struct clientInformation *customer){
112     FILE* summary = fopen("Summary.txt", "r");
113     char buffer[1024];
114     int temp = 0;
115     int line = 0;
116     int ticket = 0;
117     ticket = customer->ticket;
118     while(fgets(buffer, 1024, summary)){
119         sscanf(buffer, "%d",  &temp);
120         if(temp == ticket){
121             fclose(summary);
122             return line;
123         }
124         line++;
125     }
126     fclose(summary);
127     return line;
128 }
129
130 /*
131     findCustomerTicket uses a ticket number to find the customer in the summary
132     file then returns what line to customer is on if they exist and -1 if they don't
133 */
134 int findCustomerTicket(int ticket){
135     FILE* summary = fopen("Summary.txt", "r");
```

```
    char buffer[1024];
    int temp = 0;
    int line = 0;
    while(fgets(buffer, 1024, summary)){
        sscanf(buffer, "%d",  &temp);
        if(temp == ticket){
            fclose(summary);
            return line;
        }
        line++;
    }
    fclose(summary);
    return line;
}

/*
    createCustomer function take a customer's struct and puts their information
    into the summary text file
*/
void createCustomer(struct clientInformation *customer){
    FILE* summary = fopen("Summary.txt", "r");
    char buffer[1024];
    int temp = 0;
    int line = 0;
    int ticket = customer->ticket;
    while(fgets(buffer, 1024, summary)){
        printf("%s\n", buffer);
        sscanf(buffer, "%d",  &temp);
        if(temp == ticket){
            sscanf(buffer, "%*[^,], %[^,]",  customer->ClientName);
            sscanf(buffer, "%*[^,], %*[^,], %[^,]",  customer->DateOfBirth);
            sscanf(buffer, "%*[^,], %*[^,], %*[^,], %[^,]",  customer->Gender);
            sscanf(buffer, "%*[^,], %*[^,], %*[^,], %*[^,], %d",  &customer->GovernmentID);
            sscanf(buffer, "%*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %[^,]",  customer->DateOfTravel);
            sscanf(buffer, "%*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %d",  &customer->NumberOfTravelers);
            sscanf(buffer, "%*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %[^,]",  customer->seats);
            sscanf(buffer, "%*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %d",  &customer->server);
            sscanf(buffer, "%*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %*[^,], %[^\n]",  customer->modified);

            line++;
            break;
        }
    }

    fclose(summary);

    if(line == 0) printf("Customer was not found\n");
}
```