```
//PLEASE DO NOT DELETE. USED FOR TESTING
//JUST COPY FUNCTION STUBS TO OWN FILE TO WORK ON, CAN COMMENT THESE
OUT IF YOU NEED TO TEST

//FUNCTIONS THAT NEED TO BE IMPLEMENTED BY TEAM


#include "andrew_serverFuncs.h"


//will use customerMods.ticketNumber to search, commits modification
to summary files,
//adds note at end saying which server made modificaitons
// void modifyReservation(customerInfo customerMods, int server_name,
int socket){
//     printf("modifyReservation() called\n"); //for debugging
// }



void sendReceipt(customerInfo nextCustomer, int socket,int
server_name){
    printf("sendReceipt() called\n"); //for debugging
    //send "receipt" code to customer via tcp (client will then know
to run acceptReceipt() func)
    //then sends receipt data in form of strings (client acceptReceipt
func will create/open receipt file print)
    char stringBuffer [STRING_BUFFER_MAX];
    char bookedSeatsString [100] = " ";
    char nextSeat[8];


    strcpy(stringBuffer,"receipt");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    sprintf(stringBuffer,"\nYour Receipt From Server
%d:\n",server_name);
    send(socket,stringBuffer,sizeof(stringBuffer),0);


      sprintf(stringBuffer,"Ticket Number:
%d\n",nextCustomer.ticketNumber);
    send(socket,stringBuffer,sizeof(stringBuffer),0);


      sprintf(stringBuffer,"Your Name: %s\n",nextCustomer.fullName);
    send(socket,stringBuffer,sizeof(stringBuffer),0);

    //hardcoded dates per TA's comment that this would be acceptable
    if (nextCustomer.dayOfTravel == 1){
        strcpy(stringBuffer,"Date of Travel: May 5,2021\n");
        send(socket,stringBuffer,sizeof(stringBuffer),0);
```

```
    } else {
        strcpy(stringBuffer,"Date of Travel: May 6,2021\n");
        send(socket,stringBuffer,sizeof(stringBuffer),0);
    }

      // sprintf(stringBuffer,"Date of Travel:
%s\n",nextCustomer.dateOfTravel);
    // send(socket,stringBuffer,sizeof(stringBuffer),0);

      sprintf(stringBuffer,"Number Of Travelers:
%d\n",nextCustomer.numberOfTravelers);
    send(socket,stringBuffer,sizeof(stringBuffer),0);

      for (int i = 0; i<27;i ++){
            if (nextCustomer.bookedSeats[i] == 1) {
                snprintf(nextSeat,sizeof(int)," %d",i);
             //printf("testing seat %d",i);
                strcat(bookedSeatsString,nextSeat);
            }
      }

    printf("\nBooked seats test: %s\n",bookedSeatsString);

      sprintf(stringBuffer,"Your Booked Seats: %s
\n",bookedSeatsString);

    send(socket,stringBuffer,sizeof(stringBuffer),0);

    strcpy(stringBuffer,"THANKYOU AND HAVE A SAFE TRIP!\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

      strcpy(stringBuffer,"endreceipt");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

}



int exitProgram(int socket,availableSeats* ptr,int shm_fd){

    const int SIZE = sizeof(availableSeats)*2; //for available seats
struct in shared mem, both days
 // name of the shared memory object
    const char *name = "CS4323";

    printf("exitProgram() called\n"); //for debugging
    char stringBuffer[STRING_BUFFER_MAX];
    strcpy(stringBuffer,"\nEnter 1 to just shut down client, Enter 2
to shutdown client and server\n"); //end code to be sent to client,
client will then know to call its own exit function
    send(socket,stringBuffer,sizeof(stringBuffer),0);
```

```
    strcpy(stringBuffer,"needint"); //end code to be sent to client,
client will then know to call its own exit function
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    //sleep(1);
    int intInput; //int buffer to hold client main menu input
    recv(socket,&intInput,sizeof(int),0);
    printf("\n user entered %d\n",intInput);
    if (intInput == 1){
        printf("\n user entered %d\n",intInput);
        strcpy(stringBuffer,"end"); //end code to be sent to client,
client will then know to call its own exit function
        send(socket,stringBuffer,sizeof(stringBuffer),0);
        sleep(3); //to give time for customer to process end code
        close(socket);  //closing socket with this customer
        return 1;
    }else if (intInput == 2){
        printf("\n user entered %d\n",intInput);
        strcpy(stringBuffer,"end"); //end code to be sent to client,
client will then know to call its own exit function
        send(socket,stringBuffer,sizeof(stringBuffer),0);
        sleep(3); //to give time for customer to process end code
        close(socket);  //closing socket with this customer
        printf("\nServer about to exit\n");


        // Unmap the shared memory
        munmap(&ptr, SIZE);
        // Close the shared memory object
        close(shm_fd);
        // Delete the shared memory object
        shm_unlink(name);


        exit(0); //will need to delete this later once live server
loop in place

        return 2;
    }
}
```