```
#include "aarushi_funcs.h"
// Group I, Aarushi Singh, aarushi.singh@okstate.edu

#define STRING_BUFFER_MAX 300 //tcp string buffer, fixed size for send
and receive
struct Date getTodaysDate() {
     struct Date date;
     time_t t = time(NULL);
     struct tm tm = *localtime(&t);
     sprintf(date.today,"%d-%02d-%02d", tm.tm_year + 1900, tm.tm_mon +
1, tm.tm_mday);
     return date;
}
void writeToSummaryFile(customerInfo nextCustomer,int server_name,int
socket) { // writes to appropriate day's summary file, ticket number
will be used to search summary later on
    printf("\nwriteToSummaryFile() called...\n"); //for debugging
    dates date;

    char stringBuffer[STRING_BUFFER_MAX]; // used to send output to
server

    // used to convert bookedseats into a string to write into file
    char seat[3];
    char bookedseats[50];
    memset(bookedseats,0,strlen(bookedseats));
    char nextSeat[8];

      for (int i = 0; i<27;i ++){
            if (nextCustomer.bookedSeats[i] == 1) {
                snprintf(nextSeat,sizeof(int),"%d ",i);
             strcat(bookedseats,nextSeat);
            }
      }
    for(int i = 0; i < (strlen(bookedseats)); i++){
        if(bookedseats[i] == ' ') {
            bookedseats[i] = ',';
        }
    }

    // if day = today
    if (nextCustomer.dayOfTravel == 1) {
        // gets todays date for summary file
        char name[20];
        // struct Date date.today = getTodaysDate();
        strcpy(name,getTodaysDate().today);
        // creates summary file and writes customer info and ticket
number
        FILE * summary = fopen(name,"a");
        fprintf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: \n",nextCustomer.ticketNumber, server_name,
nextCustomer.fullName, nextCustomer.dateOfBirth, nextCustomer.gender,
nextCustomer.governmentID,
        nextCustomer.numberOfTravelers, bookedseats);
        fclose(summary);
    }

    // if day = tomorrow
    if (nextCustomer.dayOfTravel == 2) {
        // gets tomorrows date for summary file
        char name[20];
        strcpy(name,getTomorrowsDate().tomorrow);
        // creates summary file and writes customer info and ticket
number
        FILE * summary = fopen(name,"a");
        fprintf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: \n",nextCustomer.ticketNumber, server_name,
nextCustomer.fullName, nextCustomer.dateOfBirth, nextCustomer.gender,
nextCustomer.governmentID,
        nextCustomer.numberOfTravelers, bookedseats);
        fclose(summary);
    }

    // sends message to server
    strcpy(stringBuffer,"\nThe receipt was added to the summary
file!\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    stringBuffer[0] = 0;
}

int displayTicketInfo(int ticketNumber,int socket) {

    dates date;
    printf("\ndisplayTicketInfo() called.\n"); //for debugging

    // TODAY
    // gets todays date for summary file
    char name[20];
    strcpy(name,getTodaysDate().today);
    // used to send output to server
    char stringBuffer[STRING_BUFFER_MAX];
    // open summary file
    FILE * summary = fopen(name,"r");
    // struct to read file
    struct customerInfo read;
    struct customerInfo *readPTR =
    readPTR = &read;
    char filler[20];
    int server_name=0;
    char bookedseats[50];
    memset(bookedseats,0,strlen(bookedseats));

    // scan through summary file to find ticketNumber
    while(!feof(summary)) {
        fscanf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s%s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications: \n",
&readPTR->ticketNumber, &server_name, readPTR->fullName, filler,
readPTR->dateOfBirth, readPTR->gender, readPTR->governmentID,&readPTR-
>numberOfTravelers, bookedseats);
        printf("Looking for ticket...");
        if (ticketNumber == read.ticketNumber) {
            snprintf(stringBuffer, 500,"\n\nTicket Number: %d\nServer
ID: %d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment
ID: %s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications:
\n",read.ticketNumber, server_name, read.fullName, read.dateOfBirth,
read.gender, read.governmentID, read.numberOfTravelers, bookedseats);
            send(socket,stringBuffer,sizeof(stringBuffer),0);
            return 1;
        }
    }

    fclose(summary);

    printf("made it!");
    // TOMORROW
    // gets tomorrows date for summary file
    memset(name,0,strlen(name));
    strcpy(name,getTomorrowsDate().tomorrow);
    memset(bookedseats,0,strlen(bookedseats));
    // open summary file
    summary = fopen(name,"r");

    // scan through summary file to find ticketNumber
    while(!feof(summary)) {
        fscanf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s%s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications: \n",
&readPTR->ticketNumber, &server_name, readPTR->fullName, filler,
readPTR->dateOfBirth, readPTR->gender, readPTR->governmentID,&readPTR-
>numberOfTravelers, bookedseats);
        printf("Looking for ticket...");
        if (ticketNumber == read.ticketNumber) {
            snprintf(stringBuffer, 500,"\n\nTicket Number: %d\nServer
ID: %d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment
ID: %s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications:
\n",read.ticketNumber, server_name, read.fullName, read.dateOfBirth,
read.gender, read.governmentID, read.numberOfTravelers, bookedseats);
            send(socket,stringBuffer,sizeof(stringBuffer),0);
            return 1;
        }
    }
    fclose(summary);
}

customerInfo retrieveCustomersInfo(int ticketNumber) { //uses ticket
number to access sumary files and save and return customer struct
    dates date;
    printf("\nretrieveCustomersInfo called\n"); //for debugging

    // TODAY
    // gets todays date
    char name[20];
    strcpy(name,getTodaysDate().today);
    char stringBuffer[500];

    // open summary file
    FILE * summary = fopen(name,"r");
    // struct to read file
    struct customerInfo read;
    struct customerInfo *readPTR =
    readPTR = &read;
    int server_name = 0;
    char filler[20];
    char bookedseats[50];
    memset(bookedseats,0,strlen(bookedseats));

    // scan through summary file to find ticketNumber
    while(!feof(summary)) {
        fscanf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s%s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: \n", &readPTR->ticketNumber, &server_name, readPTR-
>fullName, filler, readPTR->dateOfBirth, readPTR->gender, readPTR-
>governmentID,&readPTR->numberOfTravelers, bookedseats);
        if (read.ticketNumber == ticketNumber) {
            // variables for token
            char *token;
            char tempSeat[5];

            // get the first token
            token = strtok(bookedseats, ",");

            // walk through other tokens
            while( token != NULL ) {
                sprintf(tempSeat, "%s", token );
                read.bookedSeats[atoi(tempSeat)] = 1;
                token = strtok(NULL, ",");
                printf("\nretrieveCustomersInfo called\n"); //for
debugging

            }
            read.dayOfTravel =1;
            return read;
        }
    }

    // TOMORROW
    // gets tomorrows date for summary file
    memset(name,0,strlen(name));
    strcpy(name,getTomorrowsDate().tomorrow);
    memset(bookedseats,0,strlen(bookedseats));
    // open summary file
    summary = fopen(name,"r");

    // scan through summary file to find ticketNumber
    while(!feof(summary)) {
        fscanf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s%s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: \n", &readPTR->ticketNumber, &server_name, readPTR-
>fullName, filler, readPTR->dateOfBirth, readPTR->gender, readPTR-
>governmentID,&readPTR->numberOfTravelers, bookedseats);
        if (read.ticketNumber == ticketNumber) {
            // variables for token
            char *token;
            char tempSeat[5];

            // get the first token
            token = strtok(bookedseats, ",");

            // walk through other tokens
            while( token != NULL ) {
                sprintf(tempSeat, "%s", token );
                read.bookedSeats[atoi(tempSeat)] = 1;
                token = strtok(NULL, ",");
                printf("\nretrieveCustomersInfo called\n"); //for
debugging
            }
        read.dayOfTravel = 2;
        return read;
        }
    }
}

void cancelReservation(customerInfo customerMods, int socket) {
    dates date;
    printf("\ncancelReservation() called!\n"); //for debugging

    char stringBuffer[STRING_BUFFER_MAX]; // used to send output to
server
    // variables needed to read struct
    int server_name = 0;
    char bookedseats[50];
    memset(bookedseats,0,strlen(bookedseats));
    char filler[20];

    //printf("\n%s\n",customerMods.dateOfTravel);

    printf("\n%d\n",customerMods.dayOfTravel);
    // TODAY
    if (customerMods.dayOfTravel == 1) {
        // struct to read file
        struct customerInfo read;
        struct customerInfo *readPTR =
        readPTR = &read;

        // gets todays date for summary file
        char name[20];
        strcpy(name,getTodaysDate().today);

        // creates summary file and writes customer info and ticket
number
        FILE * summary = fopen(name,"r");
        FILE * fdel = fopen("del.txt","w");

        // scan through summary file to find ticketNumber
        while(!feof(summary)) {
            fscanf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s%s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications: \n",
&readPTR->ticketNumber, &server_name, readPTR->fullName, filler,
readPTR->dateOfBirth, readPTR->gender, readPTR->governmentID,&readPTR-
>numberOfTravelers, bookedseats);
            if (customerMods.ticketNumber != read.ticketNumber) {
                fprintf(fdel, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications:
\n",read.ticketNumber, server_name, read.fullName, read.dateOfBirth,
read.gender, read.governmentID,
                read.numberOfTravelers, bookedseats);
            }
        }
        // change fdel to summary file
        fclose(summary);
        fclose(fdel);
        remove(name);
        rename("del.txt",name);
    }

    // TOMORROW
    if (customerMods.dayOfTravel == 2) {
        // struct to read file
        struct customerInfo readTomorrow;
        struct customerInfo *readTomorrowPTR =
        readTomorrowPTR = &readTomorrow;
        // gets tomorrows date for summary file
        char name[20];
        strcpy(name,getTomorrowsDate().tomorrow);

        // creates summary file and writes customer info and ticket
number
        FILE * summary = fopen(name,"r");
        FILE * fdel = fopen("del.txt","w");

        // scan through summary file to find ticketNumber
         while(!feof(summary)) {
            fscanf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s%s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications: \n",
&readTomorrowPTR->ticketNumber, &server_name,readTomorrowPTR-
>fullName, filler,readTomorrowPTR->dateOfBirth, readTomorrowPTR-
>gender, readTomorrowPTR->governmentID,&readTomorrowPTR-
>numberOfTravelers, bookedseats);
            if (customerMods.ticketNumber !=
readTomorrow.ticketNumber) {
                fprintf(fdel, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\nModifications:
\n",readTomorrow.ticketNumber, server_name, readTomorrow.fullName,
readTomorrow.dateOfBirth, readTomorrow.gender,
readTomorrow.governmentID,readTomorrow.numberOfTravelers,
bookedseats);
            }
        }
        // change fdel to summary file
        fclose(summary);
        fclose(fdel);
        remove(name);
        rename("del.txt",name);
    }
    // sends message to server
    printf("\nThe receipt was removed from the summary file!\n");

    //message to client
    strcpy(stringBuffer,"\nYour reservation has been canceled\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);

}

void modifyReservation(customerInfo customerMods, int server_name, int
socket) { // modifies reservation on summary file
    printf("\nmodifyReservation()\n"); //for debugging
    char stringBuffer[STRING_BUFFER_MAX]; // used to send output to
server

    // convert customerMods bookedseats into a string
    char seat[3];
    char bookedseatsUpdated[200];
    memset(bookedseatsUpdated,0,strlen(bookedseatsUpdated));
    for (int i = 0; i < 10; i++) {
        sprintf(seat,"%d\t",customerMods.bookedSeats[i]);
        strcat(bookedseatsUpdated,seat);
    }
    for(int i = 0; i < (strlen(bookedseatsUpdated)); i++){
        if(bookedseatsUpdated[i] == '\t') {
            bookedseatsUpdated[i] = ',';
        }
    }

    // variables to read server_name and bookedseats
    int server_name_read;
    char bookedseats[50];
    memset(bookedseats,0,strlen(bookedseats));

    // TODAY
    if (customerMods.dayOfTravel == 1) {
        // struct to read summary file
        struct customerInfo read;
        struct customerInfo *readPTR =
        readPTR = &read;

        // gets todays date for summary file
        char name[20];
        strcpy(name,getTodaysDate().today);

        // creates summary file and fdel file to hold modified info
        FILE * summary = fopen(name,"r");
        FILE * fdel = fopen("del.txt","w");

        // read through summary file
        while (fscanf(summary, "\n\nTicket Number: %d\nServer ID: 
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID: 
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
            Modifications: \n", &readPTR->ticketNumber, 
&server_name_read, readPTR->fullName, readPTR->dateOfBirth, readPTR-
>gender, readPTR->governmentID,&readPTR->numberOfTravelers, 
bookedseats) != EOF) {
            // copies all information (besides modified ticket) to 
fdel
            if (customerMods.ticketNumber != read.ticketNumber) {
                fprintf(fdel, "\n\nTicket Number: %d\nServer ID: 
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID: 
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: \n",read.ticketNumber, server_name_read, read.fullName, 
read.dateOfBirth, read.gender, read.governmentID, 
                read.numberOfTravelers, bookedseats);
            }
            // copies modified ticket to fdel
            else {
                fprintf(fdel, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: Made by server %d\n",customerMods.ticketNumber,
server_name_read, customerMods.fullName, customerMods.dateOfBirth,
customerMods.gender, customerMods.governmentID,
                customerMods.numberOfTravelers, bookedseatsUpdated,
server_name);
            }
        }
        // if dayOfTravel changes, accesses tomorrows document
        // gets tomorrows date for summary1 file
        char name1[20];
        strcpy(name1,getTomorrowsDate().tomorrow);

        // creates struct to read summary1 file
        struct customerInfo readTomorrow;
        struct customerInfo *readTomorrowPTR =
        readTomorrowPTR = &readTomorrow;

        // creates summary1 file and fdel1 file to hold modified info
        FILE * summary1 = fopen(name1,"r");
        FILE * fdel1 = fopen("del1.txt","w");

        // read through summary1 file
        while (fscanf(summary1, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
            Modifications: \n", &readTomorrowPTR->ticketNumber,
&server_name_read, readTomorrowPTR->fullName, readTomorrowPTR-
>dateOfBirth, readTomorrowPTR->gender, readTomorrowPTR-
>governmentID,&readTomorrowPTR->numberOfTravelers, bookedseats) !=
EOF) {
            // copies modified ticket to fdel
            if (customerMods.ticketNumber ==
readTomorrow.ticketNumber) {
               fprintf(fdel, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: Made by server %d\n",customerMods.ticketNumber,
server_name_read, customerMods.fullName, customerMods.dateOfBirth,
customerMods.gender, customerMods.governmentID,
               customerMods.numberOfTravelers, bookedseatsUpdated,
server_name);
            }
            // removes modified ticket from summary1
            else {
                fprintf(fdel1, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: ",readTomorrow.ticketNumber, server_name_read,
readTomorrow.fullName, readTomorrow.dateOfBirth, readTomorrow.gender,
readTomorrow.governmentID,
                readTomorrow.numberOfTravelers, bookedseats);
            }
        }

        // changes fdel file to summary file
        fclose(summary);
        fclose(fdel);
        remove(name);
        rename("del.txt",name);

        // changes fdel1 file to summary1 file
        fclose(summary1);
        fclose(fdel1);
        remove(name1);
        rename("del1.txt",name1);
    }

    // variables to read server_name and bookedseats
    server_name_read = 0;
    memset(bookedseats,0,strlen(bookedseats));

    // TOMORROW
    if (customerMods.dayOfTravel == 2) {
        // struct to read summary1 file
        struct customerInfo readTomorrow;
        struct customerInfo *readTomorrowPTR =
        readTomorrowPTR = &readTomorrow;

        // gets tomorrows date for summary1 file
        char name1[20];
        strcpy(name1,getTomorrowsDate().tomorrow);

        // creates summary1 file and fdel1 file to hold modified info
        FILE * summary1 = fopen(name1,"r");
        FILE * fdel1 = fopen("del1.txt","w");

        // read through summary1 file
        while (fscanf(summary1, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
            Modifications: \n", &readTomorrowPTR->ticketNumber,
&server_name_read, readTomorrowPTR->fullName, readTomorrowPTR-
>dateOfBirth, readTomorrowPTR->gender,
            readTomorrowPTR->governmentID,&readTomorrowPTR-
>numberOfTravelers, bookedseats) != EOF) {
            // copies all information (besides modified ticket) to
fdel1
            if (customerMods.ticketNumber !=
readTomorrow.ticketNumber) {
```

```
               fprintf(fdel1, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: \n",readTomorrow.ticketNumber, server_name_read,
readTomorrow.fullName, readTomorrow.dateOfBirth, readTomorrow.gender,
readTomorrow.governmentID,
               readTomorrow.numberOfTravelers, bookedseats);
           }
           // copies modified ticket to fdel1
           else {
               fprintf(fdel1, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: Made by server %d\n",customerMods.ticketNumber,
server_name_read, customerMods.fullName, customerMods.dateOfBirth,
customerMods.gender, customerMods.governmentID,
               customerMods.numberOfTravelers, bookedseatsUpdated,
server_name);
           }
       }
       // if dayOfTravel changes, accesses todays document
       // gets todays date for summary file
       char name[20];
       strcpy(name,getTodaysDate().today);

       // creates struct to read summary file
       struct customerInfo read;
       struct customerInfo *readPTR =
       readPTR = &read;

       // creates summary1 file and fdel1 file to hold modified info
       FILE * summary = fopen(name,"r");
       FILE * fdel = fopen("del.txt","w");

       // read through summary file
       while (fscanf(summary, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
           Modifications: \n", &readPTR->ticketNumber,
&server_name_read, readPTR->fullName, readPTR->dateOfBirth, readPTR-
>gender, readPTR->governmentID,
           &readPTR->numberOfTravelers, bookedseats) != EOF) {
           // copies modified ticket to fdel
           if (customerMods.ticketNumber == read.ticketNumber) {
              fprintf(fdel1, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: Made by server %d\n",customerMods.ticketNumber,
server_name_read, customerMods.fullName, customerMods.dateOfBirth,
customerMods.gender, customerMods.governmentID,
              customerMods.numberOfTravelers, bookedseatsUpdated,
server_name);
```

```
            }
            // removes modified ticket from summary
            else {
                fprintf(fdel, "\n\nTicket Number: %d\nServer ID:
%d\nCustomer Name: %s\nDate of Birth: %s\nGender: %s\nGovernment ID:
%s\nNumber of Travelers: %d\nSeats Booked: %s\n\
Modifications: ",read.ticketNumber, server_name_read, read.fullName,
read.dateOfBirth, read.gender,
read.governmentID,read.numberOfTravelers, bookedseats);
            }
        }

        // changes fdel file to summary file
        fclose(summary);
        fclose(fdel);
        remove(name);
        rename("del.txt",name);

        // changes fdel1 file to summary1 file
        fclose(summary1);
        fclose(fdel1);
        remove(name1);
        rename("del1.txt",name1);
    }
    // sends message to server
    strcpy(stringBuffer,"\nThe receipt was modified in the  summary
file!\n");
    send(socket,stringBuffer,sizeof(stringBuffer),0);
    stringBuffer[0] = 0;
}

struct Date getTomorrowsDate() {

    struct Date today;
    struct Date date;
    char name[20];
    strcpy(name,getTodaysDate().today);

    char *token;
    char *token1;
    char *token2;
    char *search = "-";

    token = strtok(name,search); //year
    int year = atoi(token);
    token1 = strtok(NULL, search); // month
    int month = atoi(token1);
    token2 = strtok(NULL, search); // date
    int day = atoi(token2);

    switch(month) {
        case 1:
            if (day == 31) { day = 1; month++; }
            else { day++; }
            break;
        case 2:
            if (day == 28) { day = 1; month++; }
            else { day++; }
            break;
        case 3:
            if (day == 31) { day = 1; month++; }
            else { day++; }
            break;
        case 4:
            if (day == 30) { day = 1; month++; }
            else { day++; }
            break;
        case 5:
            if (day == 31) { day = 1; month++; }
            else { day++; }
            break;
        case 6:
            if (day == 30) { day = 1; month++; }
            else { day++; }
            break;
        case 7:
            if (day == 31) { day = 1; month++; }
            else { day++; }
            break;
        case 8:
            if (day == 31) { day = 1; month++; }
            else { day++; }
            break;
        case 9:
            if (day == 30) { day = 1; month++; }
            else { day++; }
            break;
        case 10:
            if (day == 31) { day = 1; month++; }
            else { day++; }
            break;
        case 11:
            if (day == 30) { day = 1; month++; }
            else { day++; }
            break;
        case 12:
            if (day == 31) { day = 1; month++; }
            else { day++;}
            break;
    }
    sprintf(date.tomorrow,"%d-%02d-%02d", year, month, day);
    return date;
}
```