|  |  |  |
| --- | --- | --- |
| Assignment Code | : | C.L.P0005 |
| Assignment Name | : | Airline Reservations |
| Student Name | : | Le Thi Thanh Nhan |
| Time/Date | : | 3h00,22/11/2019 |

Approach

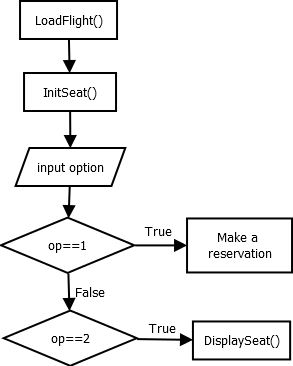
Create 2 struct:

* Struct 1: origin city, destination city, number of seats
* Struct 2: struct1, number of flight, seat[day][number of flight]

Read file and store data to 2 struct, store data about seat of each flight of each day (seat[][]).

* Make a reservation: input information and check them. If they are available then minus seat of 2 flights in input days.
* Print list of flight: print out all information of al flight.

Flowchart



Source code

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MDAY 10

typedef char string[100];

typedef struct fl

{

    string from;

    string to;

    int noseat;

} flight;

typedef struct fli

{

    flight \*DB;

    int noflight;

    int \*\*seat;

} FLIGHT;

FLIGHT LoadFlight(string fn)

{

    FLIGHT F;

    FILE \*file;

    file = fopen(fn, "r");

    fscanf(file, "%d\n", &F.noflight);

    if (file == NULL)

    {

        printf("\nError when opening the file!");

        getch();

    }

    F.DB = (flight \*)malloc(F.noflight \* sizeof(flight));

    for (int f = 0; f < F.noflight; f++)

    {

        flight flg;

        fscanf(file, "%s %s %d\n", flg.from, flg.to, &flg.noseat);

        F.DB[f] = flg;

    }

    fclose(file);

    return F;

}

void Display(FLIGHT F)

{

    for (int f = 1; f < F.noflight; f++)

    {

        flight flight = F.DB[f];

        printf("Flying from %s to %s with %d seats\n", flight.from, flight.to, flight.noseat);

    }

}

void InitSeat(FLIGHT \*F)

{

    (\*F).seat = (int \*\*)malloc(MDAY \* sizeof(int \*));

    for (int d = 1; d < MDAY; d++)

    {

        F->seat[d] = (int \*)malloc(F->noflight \* sizeof(int));

        for (int f = 0; f < F->noflight; f++)

            F->seat[d][f] = F->DB[f].noseat;

    }

}

void DisplaySeat(FLIGHT F)

{

    for (int d = 1; d < MDAY; d++)

    {

        printf("DAY: %d\n",d);

        printf("From\t\tTo\t\tSeat\n");

        for (int f=0;f<F.noflight;f++) printf("%-20s %12s %10d\n", F.DB[f].from, F.DB[f].to, F.seat[d][f]);

    }

}

int main()

{

    FLIGHT F = LoadFlight("flight.txt");

    InitSeat(&F);

    int choice;

    int dayLeave, dayBack;

    flight g;

    do

    {

        printf("\n1. Make a reservation\n");

        printf("2. Print out list of all flights\n");

        printf("3. Quit");

        printf("\nEnter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            input(&g, &dayLeave, &dayBack);

            if (dayLeave >= dayBack)

                printf("Day to come back must be later than day to leave");

            else if (dayLeave < 0 || dayLeave >= MDAY || dayBack < 0 || dayBack >= MDAY)

                printf("Invalid");

            else

            {

                int f;

                for (f = 0; f < F.noflight; f++)

                    if (!strcmp(F.DB[f].from, g.from) && !strcmp(F.DB[f].to, g.to))

                        break;

                if (f == F.noflight)

                    printf("\nWe don't have flights from %s to %s", F.DB[f].from, F.DB[f].to);

                else

                {

                    if (F.seat[dayLeave][f] < g.noseat)

                        printf("No available seat to leave");

                    else

                    {

                        int f\_back;

                        for (f\_back = 0; f\_back < F.noflight; f\_back++)

                            if (!strcmp(F.DB[f\_back].from, g.to) && !strcmp(F.DB[f\_back].to, g.from))

                                break;

                        if (F.seat[dayBack][f\_back] < g.noseat)

                            printf("No available seat to come back");

                        else

                        {

                            printf("Your reservation has been made");

                            F.seat[dayLeave][f] -= g.noseat;

                            F.seat[dayBack][f\_back] -= g.noseat;

                        }

                    }

                }

                break;

            }

        case 2:

            DisplaySeat(F);

            break;

        case 3:

            exit(0);

        }

    } while (choice >= 1 && choice <= 3);

    getch();

}

void input(flight \*g, int \*dayLeave, int \*dayBack)

{

    string name;

    printf("\nEnter name: ");

    fflush(stdin);

    gets(name);

    printf("\nFrom: ");

    fflush(stdin);

    gets(g->from);

    printf("\nTo: ");

    fflush(stdin);

    gets(g->to);

    printf("\nWhich day to leave: ");

    scanf("%d", dayLeave);

    printf("\nWhich day to come back: ");

    scanf("%d", dayBack);

    printf("\nHow many seats: ");

    scanf("%d", &(g->noseat));

}

Result

