2019 Spring - CSCI 151 — Programming Assignment

Flight Management and Booking
Part 1

Submission Deadline:
Tuesday 12 March 2019 at 23:59.

This is Part 1 of the 4-part Programming Assignment, which is worth 20% of the final grade. The 4 parts contribute to define a system for Flight Management and Booking. Please be aware that:

1. you are not allowed to discuss the assignment online;
2. you will have to submit each of the 4 parts through Moodle by the deadline;
3. you will have a 10% penalty of the total grade of the 4-part Programming Assignment for each part that you submit up to 24 hour late;
4. you will have a 25% penalty of the total grade of the 4-part Programming Assignment for each part of the assignment that does not run;
5. after all 4 parts are graded, you may be selected for live grading, in order to assess whether you understand the code and you are able to change it according to the requirements given by the examiner;
6. you will get a 0 as the total grade of the 4-part Programming Assignment if
   - you miss the submission of any of the 4 parts (submission will close 24 hours after the deadline)
   - you are selected for live grading and either you do not show up at the scheduled time or your performance does not confirm the grade of the 4 parts of the assignment;
   - you have plagiarised any of the assignment parts (either by sharing code with peers or by reusing code found online);
7. you will lose points in any of the following cases
   - you do not use appropriate code indentation;
   - you do not use comments to illustrate your code.

Do not change the names of data structures, their components, variables and functions as shown in this assignment!

1 Data Structures

Define the following data structures.

Airport information: a `struct` type `airport` with the following fields:

- a string `name` with the name of the airport;
- a string `city` with the name of the city;
- `timeZone` of type `int` to define the time zone of the city (e.g. -8 for Washington DC, 0 for London and +6 for Astana)
- `numberOfTerminals` of type `int` to define the number of terminals of the airport;
- an array `terminals` of type `char` to define the airport terminals (assume that terminals are named either with a one-digit number or an upper-case letter).

Airport list: an array `airportList` of elements of type `airport`.

Flight information: a `struct` type `flight` with the following fields:

- a string `airline` of length 2 to define the airline code;
- `number` of type `int` to define the flight number;
- `from` of type `int` to define the index of the departure airport in `airportList`;
- `to` of type `int` to define the index of the arrival airport in `airportList`;
- `hour` of type `int` to define the departure hour;
- `minute` of type `int` to define the departure minute;
- `duration` of type `int` to define the scheduled duration in minutes.

Flight schedule: an array `flightList` of elements of type `flight`.

```
Flight Departure - Astana Nazarbayev airport
Time    Flight              Destination
05:05    KC101       Moscow-Sheremetevo
06:05    SU403       Moscow-Sheremetevo
07:05    KC105       Moscow-Sheremetevo
07:30    SU502     StPetersburg-Pulkovo
12:30    KC107       Moscow-Sheremetevo
12:35    KC110       Moscow-Sheremetevo
17:15    KC112       Moscow-Sheremetevo
17:30    KC204     StPetersburg-Pulkovo
18:30    SU114       Moscow-Sheremetevo
19:30    KC116       Moscow-Sheremetevo
21:00    KC118       Moscow-Sheremetevo
```

Figure 1: Departure Information for Astana Airport

2 Functions

Define the following function:

```
void departures (const char airportName[],
                 const char airportCity[],
                 const airport airportData[],
                 const flight flightSched[]))
```

that outputs to the console departure information for a given airport `airportName` of city `airportCity` using the format in the example in Figure 1;

3 Main

Read the information in files `airports.txt` and `schedule.txt` to initialise arrays `airportList` and `flightList`, respectively.

When initialising `flightList` store only information for the flights in file `schedule.txt` such that

- departure airport and arrival airport are both in `airportList`;
- the departure hour is between 0 and 23 inclusive;

- the departure minute is between 0 and 59 inclusive;

Then call the `departures` function for each airport in `airportList`.

4 Submission Procedure

Please upload your work on Moodle as one single zipped file containing the **entire project folder**.
Deadline: **Tuesday 12 March 2019 at 23:55**.
Submissions will close on Wednesday 13 March 2019 at 23:55.