

Flight Management and Booking

Part 2

Submission Deadline:

Friday 29 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

Add to the `struct` type `flight` you defined in Part 1 of the Assignment the following fields:

- `currentDelay` of type `int` to store the possible real-time delay of the flight (to be initialised to 0 when creating elements of type `flight`);
- `depTerminal` of type `char` to store the departure terminal of the flight (to be initialised to the space character when creating elements of type `flight`);
- `depGate` of type `int` to store the departure gate assigned the flight (to be initialised to 0 when creating elements of type `flight`);

2 Functions

Modify the `departures` function you defined in Part 1 of the assignment by changing its parameters as follows:

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

where hour `hh` and `mm` are the hour and minute of the departure time according to the Greenwich Mean Time (GMT), i.e. the time in London, and by changing the function body in order to

- return the string error message "Incorrect Time" if either `hh` is not between 0 and 23 inclusive or `mm` is not between 0 and 59 inclusive;
- for each airport `airportName` of city `airportCity` (see Figures 1 and 2)
 - print the local time of `airportCity` in the header of the schedule;
and

Flight Departure - Astana Nazarbayev airport - Local Time: 07:00					
Time	Flight	Destination	Ter	Gate	Delay
07:05	KC105	Moscow-Sheremetevo	1	1	30m
07:30	SU502	StPetersburg-Pulkovo	2	2	
12:30	KC107	Moscow-Sheremetevo			720m
15:35	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 Nazarbayev Airport

- for each departing flight whose scheduled departure time (stored as `hour` and `minute` in the `struct` of type `flight`) is not earlier than hour `hh` and minute `mm` GMT (in the comparison do not consider possible delays stored as `currentDelay` in the `struct` of type `flight`), print
 - * departure time;
 - * airline code and flight number;
 - * departure city and airport;
 - * departure terminal (if not available, it is printed as a space character);
 - * departure gate (if available, i.e. if the content of `depGate` in the `struct` of type `flight` is greater than 0; if not available, it is printed as a space character);
 - * departure delay in minutes (if present, i.e. if the content of `currentDelay` in the `struct` of type `flight` is greater than 0; if not present, it is printed as a space character).

The output generated by the function has to appear according to the format in Figures 1 and 2, which are part the correct outputs of the calls of function `departures`, for information given in files `airports.txt`, `schedule.txt`, `rtFlights.txt` and `rtGates.txt`.

Flight Departure - Moscow Sheremetevo airport - Local Time: 04:00					
Time	Flight	Destination	Ter	Gate	Delay
05:40	KC102	Astana-Nazarbayev	D	14	10m
06:40	SU404	Astana-Nazarbayev	D	22	45m
07:20	SU901	StPetersburg-Pulkovo	A	19	
08:40	KC106	Astana-Nazarbayev			
12:30	KC108	Astana-Nazarbayev			
15:25	KC109	Astana-Nazarbayev			
18:05	SU113	Astana-Nazarbayev			
19:00	KC115	Astana-Nazarbayev			
20:00	KC117	Astana-Nazarbayev			
22:00	KC119	Astana-Nazarbayev			

Figure 2: Departure Information for Moscow Sheremetevo Airport

3 Main

Read the information

- in file `airports.txt` to initialise array `airportList`;
- in files `schedule.txt`, `rtFlights.txt` and `rtGates.txt` to initialise array `flightList`.

Please note that the contents of files `airports.txt` and `schedule.txt` are partly different from the ones in Part 1 of the Assignment.

As in Part 1, 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` with time 01:00 GMT. For example, the output for Nazarbayev Airport in Astana at time 01:00 GMT, i.e. time 07:00 in Astana, is given in Figure 1 and the output for Sheremetevo Airport in Moscow at time 01:00 GMT, i.e. time 04:00 in Moscow, is given in Figure 2.

4 Submission Procedure

Please upload your work on Moodle as one single zipped file containing the **entire project folder**.

Deadline: **Tuesday 26 March 2019 at 23:55.**

Submissions will close on Wednesday 27 March 2019 at 23:55.