PROGRAMMING TECHNIQUES, A.A. 2021/2022
Laboratory 6

Exercise 1 is a home assignment that can be optionally submitted for evaluation, to obtain the exam's bonus. Deadline is **13/5/2022, 11:59 pm**, and it will be the same for laboratories 4, 5 and 6. The home assignments of Lab4+Lab5+Lab6 need to be submitted in one shot by the given deadline, following the instructions that are provided in the Portale della Didattica (see the document: Instructions for submission of assigments.pdf)

Objectives
- Solve iterative text processing and verification/selection problems, using arrays and matrices (*C3b- Problem solving with arrays*)

Technical content
- I/O basics
- Functions
- Conditional and iterative problems
- Operations with arrays (of int, float and char)
- String processing functions

Exercise 1. (THIS EXERCISE IS TO BE SUBMITTED FOR THE EXAM BONUS)
Category: selection problems

Transport company

An urban transport company tracks the trips made by its vehicles in a log file (text file named `log.txt`). The file is organized as follows:
• in the first line, a positive integer indicating the number of lines that will follow (no more than 1000)
• in the next lines, the information about the trips made by the vehicles, one per line with the format:

```
<code_route> <departure> <destination> <date> <time_departure> <time_arrival> <delay>
```

`<code_route>`,`<departure>`,`<destination>` are strings of 30 characters at most, respectively representing the route code, the departure and the destination of the trip;
`<date>` is the date of the trip in the format `yyyy/mm/dd`;
`<time_departure>`,`<time_arrival>` are the `hh:mm:ss` times of departure and arrival of the trip;
`<delay>` is an integer >=0 representing the delay accumulated during the trip. Assume that the delay never impacts on the date of a trip.

Example of file `log.txt`:
```
6
GTT001 Braccini Porta_Nuova 2018/10/10 18:50:00 19:07:25 1
GTT001 Braccini Porta_Nuova 2018/12/10 19:50:00 20:06:00 1
GTT002 Politecnico XVIII_Dicembre 2018/10/10 10:01:23 10:12:08 4
GTT003 Einaudi Cso_Trapani 2018/09/10 14:11:23 14:38:23 2
GTT004 Marmolada Sebastopoli 2018/11/10 00:01:02 00:12:00 3
GTT002 Politecnico Piazza_Statuto 2018/11/10 23:11:59 23:20:07 0
```

After reading the file and acquiring its content in an appropriate data structure, the C program should provide a menu to the user, to choose one of the following operations:
1. list all rides departed within a certain interval of dates
2. list all the trips starting from a given departure
3. list all the trips with a given destination
4. list all trips that reached their destination late, within a given interval of dates
5. list the overall delay accumulated by the trips that are identified by a given route code
6. terminate the program

To select one of the available operations, the user is asked to enter a command from keyboard. A command consists of a specific word, one per each operation (respectively, "`date`", "`departure`", "`destination`", "`delay`", "`tot-delay`" or "`end`"), eventually followed by the additional data required by the operation, all in the same line. For example, "`date`" should be followed by two dates in the `yyyy/mm/dd` format, "`departure`" should be followed by the name of the departure point, etc.

To handle the selection from the menu, use a strategy similar to the example reported in slide 73 of *C3b-Problem solving with arrays*. More specifically, you should define a new `enum t_command` type, including the symbols `c_date`, `c_departure`, `c_destination`, `c_delay`, `c_tot_delay`, `c_end`. Encode each command with the corresponding `t_command` symbol, so that you can easily handle the menu selection with a `switch-case` construct.

Suggestions:
- implement a `readCommand` function that acquires the first word of a command from keyboard and returns the corresponding `t_command` value
- implement a `menuWord` function that takes as parameters the data structure where the trips are stored and the number of stored trips and handles with a menu the acquisition of the commands (initial word with corresponding additional information), as well as the call to the appropriate functions implementing the corresponding operations.

Exercise 2.

Category: text processing with strings

Occurrence of words

Write a program able to identify in a text file words that contain a given sequence of characters. So-called "words" in the text contain only alphanumeric characters (they can be identified by the `isalnum` function from `ctype.h`) and are separated by spaces or punctuation marks (respectively identified by the `isspace` and `ispunct` functions).

The program should receive in input:
- The file `sequences.txt`. The file reports in the first line an integer value representing the total number of given sequences (no more than 20), and in the following lines, one per line, the sequences to be identified. Each sequence is no more than 5 characters long. You should not differentiate between uppercase and lowercase characters.
- The file `text.txt`. The file contains a generic text, of unknown number of lines. You can assume that each line is no more than 200 characters long and that each word is no more than 25 characters long.

The program should print on the screen, per each of the given sequences, the list of words in the text that contain such sequence, with their respective position in the file. The position should be computed in terms of number of words from the beginning of the file (that is: 1 for the first word of the text, 2 for the second, etc.) For the sake of this exercise, you should identify and visualize only a maximum of 10 words per each given sequence.

Example

file `sequences.txt`:

```
4
no
Al
per
s
```

file `text.txt`:

```
Non sempre si capisce un esercizio alla prima lettura, ma prestando
attenzione al testo e all’esempio non dovrebbe essere impossibile
scrivere codice funzionante nonostante i dubbi iniziali. Se ancora non
si capisce, allora basta chiedere all’esercitatore di turno.
```

Sequence `no` is contained in:

`Non` (position 1)

`non` (position 18)

`nonostante` (position 25)

`non` (position 31)

`turno` (position 40).

…

…