04JCJLZ - COMPUTER SCIENCES - 2015/2016

Laboratory 11

Objectives

- To write programs using file operations and advanced use of structures

Technical content

- Consolidating read data from files
- Defining data structures
- *typedef* and .

Preferably to be solved in the laboratory

Exercise 1. Write a program that allows the storage of a set of coordinates in the Cartesian coordinate system. Define in the program a data structure containing two fields as follows:

```
struct coordinate
{
     int x;
     int y;
};
```

The program shall acquire from the keyboard the coordinates *x* and *y* of 4 points, corresponding to the points of a path and then, the program manipulates the received data according to a parameter received from the command line:

- Parameter *"—m"* : computes and prints on the screen the length of the path composed by the 4 segments.
- Parameter *"—a"* : computes and prints on the screen the minimum length between the given coordinates.

Exercise 2. Write a program for the management of a book of contacts of maximum 100 contacts.

The program shall allow the storage of following data in a structure:

```
typedef struct names {
     char name[20];    /*Name*/
     char surname[20]; /*Surname*/
     char number[20];  /*Phone Number*/
     char mobile[20];  /*Mobile Number*/
} Names;
```

The program shall allow the user to perform, by means of a menu, the input of a new contact (allowing eventual duplicates of name and surname, but shall give a warning to the user in case of duplicates and shall ask to confirm the input), and displays the complete list.

Exercise 3. [1]Write a C program that reads the contents of a file whose name is acquired from the command line. The number of rows in the file is up to *80* and each row of the file contains the following fields:

<subject> <prof name> <prof surname> <period> <credits> <% pass rate>

Each field of the row is composed of up to *20* characters and contains no spaces. The program must print on the screen the following information:

a) the name of the subject that assigns the maximum credits;
b) for each academic period (considering maximum *4*), the name of the subject that is most difficult to pass.

Afterwards, the program will ask the user to enter a professor’s last name (maximum *20* characters) and print the following information:

c) the sum of the credits assigned with the courses taught by this professor;
d) the average pass rate of the courses taught by this professor.

Further insight: for points a) and b), consider the case that two or more subjects are assigned with the maximum credits and that two or more subjects in the same period have the lowest pass rate. The program must output the list of all the identified subjects.

The following is a possible example of the input file:
HorseRiding Donato Cavallo 1 5 50
Rowing Remo Controcorrente 2 4 70
Speed Tina Svelta 1 10 80

To be solved at home

Exercise 4. Write a C program that reads from a file (which name is acquired as the first parameter in the command line) the information about arrivals and departures in a train station. For each line, the file contains the following information (each field contains maximum 20 characters and no spaces):

<departure_station> <departing_time> <destination_station> <arriving_time>

[1] This exercise will be solved using a multimedia format, and its solution will be provided in the course site during the following weeks.

The program receives a second parameter from the command line that indicates the name of a city; then, the program should calculate and print the number of trains arriving and departing from the given city.

Exercise 5. Write a C program for the management of an athletic contest. The program shall:

a. read from a text file named *contest.txt* a non-ordered sequence of names with associated score. The names does not contain spaces and the score is an integer; it is assumed that there are at most 100 athletes;
b. display on the screen the sequence of names in alphabetic order, with associated score;
c. simultaneously, write the output into a text file named *contest_ord.txt*, of the same ordered data;
d. display on the screen the names of the best three positions, without reordering the sequence.

Remember that the first classified is the athlete with the highest score, the second is the one with the highest score lower than the first one, and so on (in general, each time it should be searched for the maximum value, this must be lower than the maximum value found previously).

Note: it is suggested to perform an ordered insertion into an array of *struct*, as seen in the previous laboratory exercises.

Exercise 6. Write a C program for the management of the exams of this course. The commands that the programs shall handle are the following:

- I <Surname_Name> <StudentID> <Score>
  to insert the student's data.
- D <Surname_Name>
  to delete the student's data given the name.
- D <StudentID>
- to delete the student's data given the ID.
- V <Surname_Name>
  to view the student's data given the name.
- V <StudentID>
  to view the student's data given the ID.
- P
  to print the whole data-base.
- L <Filename>
  to load the data-base (already ordered) from a file, which name is given after the command *L*.
- S <Filename>
  to save the data-base into a file, which name is provided after the command.

Use an array of *struct* ordered by the field <Surname_Name>.

Note: the commands must be provided (and the program must accept them) according to the format specification, without redundant messages. For example, to insert the data of a student:

I Michael Andrew Fox 123456 30

Therefore, concerning D (delete) and V (view) commands, the program must deduce (from the parameter) whether the user has entered a name or a student ID.

Assume that the <Surname_Name> fields do not contain digits, the <StudentID> are only numeric, and the input is always correct.