

## Lab. 4 Input/Output to Text Files; Dictionaries.

Do the exercises below in the Spyder IDE. Make sure the files and the programs are in the same working directory.

### 1. Writing to a text File

- a) Implement function below to write, into a file with the specified `fname`, all elements of integer vector `v`, in separate lines. The file should start with the sentence “**The following integers are the k elements of a vector**” where `k` is the number of elements of the vector.

- `def write_vector(V, fname)`

- b) Implement function below, similar to the previous one, but writing into the file all elements of matrix `Mat`, in separate lines, row by row. The file should start with the sentence “**The following integers are the m \* n elements of a matrix**” where `m` and `n` are, respectively the number of rows and columns of the matrix.

- `def write_matrix(Mat, filename)`

### 2. Reading from a text File

Implement functions below, that return, respectively, a vector and a matrix from files with name `fname`, with the same format of those specified in the previous question.

- `def read_vector(filename)`
- `def M = read_matrix(filename)`

Test your functions with the files obtained in the previous question.

### 3. Input a Table of dictionaries

Specify function with signature

```
def read_table(fname):
```

that returns a table (a list of dictionaries), `T`, with the data stored in file with `fname`. The first line of the file indicates the title of the table, the second the names of the fields, separated by semi-colons (“;”), and the subsequent lines the table data (also separated by semi-colons (“;”).

Test your function with file `students.txt`, available in the website. Note that the grades can be either a real number greater or equal than 9.5, a “freq” indicating that the student can do an exam, and “rep” indicating the student failed.

### 4. Process a Table of Dictionaries

For table `T`, obtained in the previous question, specify functions to answer the following question:

- a) How many students failed already in the course?

```
def n_failed(T)
```

- b) What is the average grade of the students that have already approval the course? And the best and worse grade obtained by the students approved. The function with signature

```
def statistics(T)
```

should returns a dictionary with these 3 values (average, maximum and minimum grades) with keys ‘mean’, ‘highest’ and ‘lowest’.

- c) Given a table `T` (as before), return a table with the numbers and names of the students with grades in the interval `[min_grade...max_grade]`.

```
def students_range(T, min_grade, max_grade)
```

- d) Given a table `T` (as before), return another table `A`, with the grades of the already approved students rounded to an integer (between 10 and 20).

```
def students_approved(T)
```

## 5. Writing a Dictionary

- a) Write a function that writes a table into a file, with a first line containing a title (given as argument), the second line containing the keys of the different fields, separated by semi-colons (“;”), and the subsequent lines with the data of the table, also separated by semi-colons (“;”). Test your function with table **A**, obtained in question 2.d), writing it to file **approved.txt**.

```
def write_table(fname, table, title)
```

- b) Use again the function to write into the file **students\_x.txt** all the students from table **T** above, that have their full name started with a given name.

```
def write_students_with_name(T, name)
```

## 6. Reshaping a Table Dictionary

- a) Reorganize table **T** above, as a dictionary of dictionaries, such that the key passed as argument is used not only in each student record, but also as a key to that student in the table. The key is assumed to be a number

```
def reshape(T, key)
```

## 7. XML text files

- a) Specify a function that writes a table of students into an xml text file. Test your function with tables **T** and **A** of questions 1, and 2.d) respectively, writing it to files **xml\_students.txt**, and **xml\_approved.txt**. Assume title “Computing Literacy”.

```
def write_table(fname, table, course)
```

**Note:** The xml format of the file should be as follows, where ... should be filled by the appropriate values. The students between tags <st> and </st>, should be as many as there are in the table.

```
<grades>
  <course> ... </course>
  <students>
    <student> <number> ... </number> <name>...</name> <grade>...</grade> </student>
    <student> <number> ... </number> <name>...</name> <grade>...</grade> </student>
    .....
    <student> <number> ... </number> <name>...</name> <grade>...</grade> </student>
</grades>
```

- a) Specify a function that reads a table (and a course) of students from an xml text file with the format above. Test your function with the files written in the previous question.