Laboratory 1

Exercise 1 (adapted from Computer Sciences exam 03/09/2012)

Write a program able to handle the scores of an artistic gymnastic event. The event scores are contained in a file whose name is passed from the command line. Every line of the file contains: the competitors name and surname, competitors nationality, and the assigned evaluations provided by 5 judges. For example, the file (`score.txt`) contains:

```
Donald Duck ITA 9.3 8.9 9.7 9.7 9.8
Mickey Mouse ITA 9.0 9.0 9.0 9.2 9.5
Bugs Bunny USA 8.4 8.7 8.5 8.6 9.0
Daffy Duck RUS 8.3 8.8 9.5 9.6 9.0
Charlie Brown GRB 8.2 8.9 8.9 8.6 9.3
```

Assumptions:

- The total number of records (lines) is not known
- The competitors name and surname do not contain spaces
- There are always 5 evaluations for every competitor, and these numbers are separated by a space

The program should show:

- The final ranking for the best 3 competitors: consider that for computing the final records, the highest and lowest evaluations are ignored and the score is determined by the sum of the remaining 3 values.
- The best country: the one that obtained the best score considering the sum of all the competitors of this country (always ignoring for every competitor the highest and lower evaluations).

For the provided file, the program output should be

```
final ranking:
1: Donald Duck -- Score: 28.7
2: Daffy Duck  Score: 27.3
3: Mickey Mouse  Score: 27.2

Best Country:
ITA  Total score: 55.9
```

Exercise 2 (adapted from Computer Sciences exam 23/06/2017)

You are asked to implement a program for managing the database of a city public transportation center. The information is stored in a file whose name is passed as command line argument. Each line in the file contains: the ID number of the public transport bus, the number of route the bus serves, the geometric coordinates in meters (abscissa and ordinate, i.e., x-axis and y-axis) and the time in seconds (all the time values belong to the same day) when the bus is checked. For example, the file can contain:

```
2187 13 10 1003 18000
3002 4 5000 5 18100
2187 13 100 2030 18500
3002 4 5000 1100 18600
2187 13 300 3300 19200
3002 4 5000 2200 19200
1976 4  5000 5 18600
1976 4  5000 1100 19600
1976 4  5000 2200 20100
```

We make the following assumptions:

- All the file can be loaded in memory
- Each bus serves only one line
- Multiple buses can serve the same line

The program receives the following parameters from the command line: 1) the name of the file containing the database and, 2) a flag, followed by an additional parameter.

- If the flag is ’-b’, the next parameter is a busId. The program should print the total distance traveled by the given bus
- If the flag is ’-b’, the next parameter is a lineId. The program should print the average speed of buses traveling on the line

For example:

```
$> python lab_1_2.py lab_1_2.txt -b 1976
1976 - Total Distance: 2195.0

$> python lab_1_2.py lab_1_2.txt -l 4
4 - Avg Speed: 1.6884615384615385
```