

COURSE OF COMPUTER SCIENCE *LABORATORY PRACTICE n. 10*

Exercise 1 (1 point):

A text file contains an unknown number of integer values, separated by one or more spaces or by a “new line” character. Write down a program which, by reading such a file only once (the file name should be asked to the user), computes and displays on screen: the number of values contained in the file, their sum, as well as the maximum and minimum values among them.

Example: Assume the following file contents:

```
23 47      11
-3      12  -24  9  0
      8
```

Then, the program should display the following messages:

```
Number of values: 9
Overall sum: 83
Maximum value: 47
Minimum value: -24
```

Exercise 2 (2 points):

Write down a program which loads the name of a text file from the keyboard and, by reading it only once, computes and prints out the following statistics on the text there contained:

- Total number of rows
- Total number of characters
- Number of digits (characters ‘0’, ‘1’, ..., ‘9’)
- Number of capital letters (‘A’, ‘B’, ..., ‘Z’)
- Number of small letters (‘a’, ‘b’, ..., ‘z’)
- Number of spacing characters (‘ ’, ‘\t’, ‘\n’, etc.)
- Number of punctuation characters (‘.’, ‘,’, ‘:’, ‘;’, ‘?’, ‘!’)
- Number of characters not included in the previous categories

Exercise 3 (2 points):

Write down a Python program in order to “justify” a given text, i.e., to rewrite it in such a way that all its lines have a fixed standard length (let `LENGTH`, a predefined constant, be the value of such an amount), with all the words inside the same line being separated by (approximately) the same number of spaces.

The text is contained in a file and it is composed by an unknown number of lines. The justified text must be written into another file: the two file names should be received by the program as command line parameters. Assume that:

- Every row of the input file has a length which is smaller than `LENGTH`.
- Every row of the input file contains at least two words.

- All the words belonging to the same row of the input file are separated by exactly one space.

Example: Assume `LENGTH=60` and the following contents for the input file:

```
This text is just an example.
This row is short.
This row is much longer than the previous ones.
```

Then, the output file contents should be:

```
This      text      is      just      an      example.
This      row      is      short.
This  row  is  much  longer  than  the  previous  ones.
```

Exercise 4 (2 points):

A university secretary makes available to a professor a text file (named “`students.txt`”), where, for each row, the name and the matricula of a student is reported. At the end of his course, the teacher provides another file (“`marks.txt`”) in which he writes, for each line, the matricula of a student and the *two* marks the student obtained. No correspondence between the rows of the two files can be assumed. The total number of students is unknown, but limited to 200.

Write a program which, by reading such files only once, prints on screen, for each student, his name and the final mark he got. The final mark is:

- `FAIL` if at least one of the marks is less than 15 or their average is less than 18.
- The average of the two marks (approximated to the first larger integer value) otherwise.

Example: Assume the following contents for the two input files:

| <code>students.txt</code> | <code>marks.txt</code> |
|---------------------------|------------------------|
| MickeyMouse 1575 | 7302 28 30 |
| DonaldDuck 7302 | 1575 14 26 |
| Goofy 4529 | 4529 23 16 |

Then, the following messages should be displayed (the order is not relevant):

```
Goofy      20
MickeyMouse FAIL
DonaldDuck 29
```

Exercise 5 (2 points):

A bi-dimensional matrix with unknown sides (however limited to 20 rows and 20 columns) represents a crossword puzzle. Each element of this matrix corresponds to one cell of the puzzle, and it may assume only values 0 and 1, for a white or black cell, respectively.

Write a Python program able to:

- Read from the keyboard the name of the file in which the matrix is stored.
- Load the matrix from such file.
- Draw the crossword scheme, according to the following example.

Example: let the matrix stored in the file be:

```
10100100
```

00001000
10010010
11001100

Then, the program must display the following scheme:

```
-----
| * * * * * |   | * * * * * |   |   |   | * * * * * |   |   |   |
| * * * * * |   | * * * * * |   |   |   | * * * * * |   |   |   |
| * * * * * |   | * * * * * |   |   |   | * * * * * |   |   |   |
| * * * * * |   | * * * * * |   |   |   | * * * * * |   |   |   |
-----
|   |   |   |   |   | * * * * * |   |   |   |   |   |   |
|   |   |   |   |   | * * * * * |   |   |   |   |   |   |
|   |   |   |   |   | * * * * * |   |   |   |   |   |   |
|   |   |   |   |   | * * * * * |   |   |   |   |   |   |
-----
| * * * * * |   |   |   | * * * * * |   |   |   | * * * * * |   |   |   |
| * * * * * |   |   |   | * * * * * |   |   |   | * * * * * |   |   |   |
| * * * * * |   |   |   | * * * * * |   |   |   | * * * * * |   |   |   |
| * * * * * |   |   |   | * * * * * |   |   |   | * * * * * |   |   |   |
-----
| * * * * * | * * * * * |   |   |   | * * * * * | * * * * * |   |   |   |
| * * * * * | * * * * * |   |   |   | * * * * * | * * * * * |   |   |   |
| * * * * * | * * * * * |   |   |   | * * * * * | * * * * * |   |   |   |
| * * * * * | * * * * * |   |   |   | * * * * * | * * * * * |   |   |   |
-----
```