Lab. 4 Text Processing; Input/Output to Text Files

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

1. Text Processing

Create a sentence in a string variable, for example

"**This string (created for testing), has 70 characters, 17 being vowels.**"

and use it to test the following functions that you should implement:

- **function nc = n_chars(str)**
  - **nc** the number of characters in the string str
- **function nd = n_digits(str)**
  - **nd** the number of digits in the string str
- **function nv = n_vowels(str)**
  - **nv** is the number of vowels in the string str
- **function nw = n_words(str)**
  - **nw** the number of words in the string str (a word is a sequence of alpha chars)
- **function ni = n_integers(str)**
  - **ni** the number of integers in the string str (an integer is a sequence of digits)

2. Number of Substrings

a) Implement the following functions, using no predefined MATLAB string functions

- **function n = n_occurs(sub, str)**
- **function n = n_occurs_no_over(sub, str)**

that return the number of occurrences of the string **sub** in string **str**, allowing or not overlapping. For example given strings **str = "arara"** and **sub = "ara"**, function **n_occurs** should return 1, whereas function **n_occurs_over** should return 2.

b) Implement alternative versions of the functions using no predefined MATLAB string functions.

3. 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.

- **function 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..

- **function write_matrix(Mat, filename)**

4. Reading from a text File

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

- **function V = read_vector(filename)**
- **function M = read_matrix(filename)**

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