VG101 — Introduction to Computer and Programming

Assignment 1

Manuel — UM-JI (Fall 2018)

- MATLAB: write each exercise in a different file
- C/C++: use the provided assignment template
- Include simple comments in the code
- If applicable, split the code over several functions
- Extensively test your code and impove it
- Write a single README file per assignment
- Zip all the files and upload the archive on Canvas

*Exercises preceded by a * are mandatory.Any student not completing all of them, or submitting a work that cannot compile or be interpreted will automatically receive a -5% penalty on the final course grade.*

JOJ Online Judge

Exercises 2 and 6 can be tested on JOJ Online Judge.

Important reminders regarding the Online Judge (OJ):

- For each exercise save all the files, without any folder structure, into a `.tar` archive;
- Strictly stick to the input and output formats provided in the specifications;
- The OJ only checks the correctness of the code not its quality;
- For feedbacks on the quality, submit the code as part of the assignment and include the OJ score as well as the failed cases in the README file;

* **Ex. 1** — *Writing proper documentation*

Search online what a README file is and what it should contain.

Specifications.

- Answer in the README file
- Provide clear explanations

* **Ex. 2** — *Basic MATLAB knowledge*

Let $x = \begin{pmatrix} 3 \\ 2 \\ 6 \\ 8 \end{pmatrix}$ and $y = \begin{pmatrix} 4 \\ 1 \\ 3 \\ 5 \end{pmatrix}$

1. Define x and y in MATLAB.
2. Add the sum of the elements in x “at the end of” y.
3. Raise each element of x to the power specified by the corresponding element in y.
4. Divide each element of y by the corresponding element in x.
5. Multiply each element in x by the corresponding element in y, and store the result in a variable z.
6. Add up the elements in z and assign the result to a variable w.
7. Compute $x' * y - w$ and explain the result.

Specifications.

- For each question display all the variables in their order of appearance in the question
- At the end of each step reset x and y to their initial value from question 1

Ex. 3 — *Simple scripting in MATLAB*
Write a simple MATLAB script which converts durations from seconds into days. The input should be represented as a vector of 1000 durations from 60 to 12345987 seconds.

Ex. 4 — *Basic scripting*
The length of a competitive running track is 400 m. Write a MATLAB script which converts a distance into a number of laps. When the number of laps is not an integer also return the number of meters remaining in order to complete one more lap.
e.g. For 1100m the script should return: 2 and 100 (2 laps and 100 m to complete 3 laps).

Specifications.

- The script should display the result
- E.g. for 1100m show 2 and 100, meaning 2 laps and 100 m to complete 3 laps

* **Ex. 5** — *Algorithms*
Around 240 BC Eratosthenes calculated the circumference of the Earth using basic mathematics and without leaving Egypt.

1. In the README file explain his method, and write a clear algorithm describing it
2. Write a MATLAB script which calculates and displays:
   a) The circumference of the Earth
   b) The radius of the Earth

* **Ex. 6** — *Vectors*
In Great Britain and Ireland human body weight is often measured in stones. A stone is defined as 14 pounds or 6.35 kg. Write a MATLAB script converting from stones to pounds, from pounds to kg and from kg to stones.

Specifications.

- Input format: two lines containing (i) a list of values and (ii) the expected conversion (e.g. `[ 3.14 6.28 9.42 ]` on the first line and `stones to pounds` on the second one
- Output format: a vector containing the result of the conversion
- Use vectors to store the values

Ex. 7 — *Compact coding*
For each question use a single command line in MATLAB to determine and print:

1. The number of primes between 1 and 100000.
2. The product of 5 random numbers in the range 1 to 10.