VG101 — Introduction to Computer and Programming

Homework 3

Manuel — UM-JI (Summer 2019)

- 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
- Archive the files (`*.{zip|tar}`) and upload 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 be deducted 1 mark on the final course grade.*

JOJ Online Judge

Exercises 1, 3, and 5 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** — *Accurate calculations*

When an ancient Indian Brahmin invented the game of Chess he showed it to his king who was really impressed. He was so pleased that he allowed him to chose his own reward. The inventor replied that he wanted one grain of wheat on the first square of the chess board, two on the second, four on the third, eight on the fourth, and so forth...It took more than a week to the treasurer to calculate the amount a wheat required. Write a MATLAB script to help him determine how many grains of wheat had to be exactly given to the creator of Chess. The story ends with the creator of Chess becoming the new king.

Output format.

🟩 • Output format: a single line displaying the result

Ex. 2 — *Ploting*

Plot a simple house and a car using basic geometric shapes such as rectangles, trapeziums, circles, and triangles.

* **Ex. 3** — *Structures*

The following table summarizes a wardrobe inventory. Create an appropriate MATLAB structure to represent the data, and write a script to determine (i) which item (Type+Color) is in the largest quantity and (ii) how old are the items in average – age in years, rounded down.

Type	Color	Quantity	Bought
Jumpers	Blue	2	04/2005
	Brown	3	02/2013
	Green	5	01/2015
Trousers	Black	3	06/2012
	Grey	2	04/2011
	White	1	12/2013
T-shirts	Blue	1	05/2010
	Green	2	09/2014
	Red	3	01/2012
	White	2	03/2008
	Yellow	1	11/2012

Specifications.

- Output format: two lines, the first one containing a string of the form “type color” (e.g. `Jumper Green`) and the second one an integer representing the rounded-down average age
- Current year is to be taken as 2019

Ex. 4 — *Input and output*

Write a MATLAB function which takes as input an integer n and dumps in a file all the multiples $n \times i$, $0 \leq i \leq 10$. Respect the following format (do not output the dots, and the line numbers):

```
23 x 0 = 0
23 x 1 = 23
23 x 2 = 46
...
```

* **Ex. 5** — *Algorithm, function, conditional statements, and loops*

Given a continuous function f over an interval $[a, b]$ such that $\text{sign}(f(a)) \neq \text{sign}(f(b))$ find $r \in [a; b]$ such that $f(r) = 0$. The bisection method consists in dividing the interval $[a; b]$ into two sub-intervals $[a; c]$ and $[c; b]$ of equal size. Then either $f(a)$ and $f(c)$ or $f(c)$ and $f(b)$ will have different signs. In case $c = r$ we stop and return c, otherwise the process is repeated over the interval where the sign changes. The process of narrowing down the interval will only end when the error is smaller than a bound specified by the user.

1. Write a clear algorithm describing the bisection method
2. Implement the previous algorithm using a MATLAB function

Note: the degree of accuracy should be at least 0.001 (strictly positive and less than 0.001).

Specifications.

- Input format: two lines, the fist one containing a function (e.g. `@(x) x^2-3*x+1`), and the second one showing an interval with a root (e.g. `[0 1]`)
- Output format: one lines displaying a number whose degree of accuracy should be at least 0.001 if the root has more than three digits after the decimal point

Ex. 6 — *Input and output*

Pascal's triangle is a triangular array composed of the binomial coefficients. Write a MATLAB function taking as input an integer n and which outputs n lines of Pascal's triangle in a text file.

Hint: either generate it using the fact that each number in the triangle is the sum of the two numbers directly above it or using the functions `pascal` `diag`*, and* `rot90`*.*

Output for the case $n = 6$

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5   10  10  5   1
```