VG101 — Introduction to Computer and Programming

Assignment 2 (17/10/2016)

Manuel — UM-JI (Fall 2016)

- Write each exercise in a different file
- Include simple comments in the code
- If applicable, split the implementation over several functions
- Write a single README file per assignment
- Archive all the files in a `zip` file and upload it onto Sakai

Ex. 1 — *Algorithms and basic loops*

In the Gregorian calendar a regular year is composed of 365 days, and a "leap year" of 366 days, the 29th of February being added in order to reflect more precisely the orbital period of the Earth around the Sun. Determining whether a year is a leap year or not in the Gregorian calendar is done as follows: if a year is not divisible by 4 it's a regular year; if it's divisible by 100 then it's not a leap year, unless it's also divisible by 400.

Write a MATLAB script returning whether a given year is a leap year or not. The user should be prompted for a year until he enters a valid one.

Ex. 2 — *Algorithms and loops*

A Pythagorean prime is an odd prime number that can be written as the sum of two squares. Such primes are of the form $p = 4n + 1$, for some integer n. Write a MATLAB script that (i) reads a number from the keyboard, then (ii) finds the next Pythagorean prime and (iii) returns the two corresponding squares.

Ex. 3 — *Recursion*

Chapter 5 describes an algorithm to read numbers digit by digit (slides 5.24 and 5.25). Instead we now want to read them as proper numbers, e.g. 452 should be "four hundred and fifty two". Write some MATLAB function/subfunctions such that an integer input less than 999,999,999 is returned as words.

Ex. 4 — *Mathematical functions, loops, and recursion*

Given a continuous function f over an interval $[x_0, x_1]$ such that $\mathrm{sign}(f(x_0)) \neq \mathrm{sign}(f(x_1))$ find $r \in [x_0; x_1]$ such that $f(r) = 0$. The secant method is defined through the following recurrence relation

$$x_n = \frac{x_{n-2} f(x_{n-1}) - x_{n-1} f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})}$$

Write two MATLAB functions, one iterative and one recursive, taking as input a mathematical function, an interval containing a root, and a precision. They should return the root of the function in the interval provided in the input.

Ex. 5 — *Control statements*

A positive integer n is an Armstrong number if the sum of the i-th power of each of its digits is n itself, with i the number of digits in n. For instance 153 is an Armstrong number for (i) it has three digits and (ii) $1^3 + 5^3 + 3^3 = 153$. Similarly 1 is also an Armstrong number since (i) 1 has 1 digit and (ii) $1 = 1^1$. Write a MATLAB function which given a number n returns the next Armstrong number or n if n is an Armstrong number.