Unit 3

Practice IV

Algorithms

Goal of the practice

Know how to use invocation of algorithms and what is the purpose of them. Use predefined functions and create new functions to solve problems. Notion of re-usability.
Understand the difference between parameters and arguments.
Understand the definition of the scopes of variables.

1) What are the advantages and disadvantages of modular programming. Explain in words how to model with modules a problem where you have to read from the keyboard a series of numbers (which can be stored in a list) and display the 3 major numbers and 3 minor ones.

2) Count the number of words, separated by one or more spaces, from a telegram ending in point. It can be assumed that the user enters character by character the telegram or the complete sequence, which is more comfortable to propose a solution.

3) Given the following algorithms:

```
algorithm a1
   var Number : x
   x ← 10
   print (x * 5)
   x ← a2(x*2)
   x ← x * 2
   a2(x)
end algorithm

algorithm a2(Number : x) : Number
   print (x)
   x ← x / 2
   print (x)
   return x
end algorithm
```

Identify the scopes of each variable. Track the execution of a1 indicating what is displayed by console. Idem for the execution (independently) of the algorithm a2.

4) Develop an algorithm for a dice game. The player must bet on a number between 1 and 6 (keyboard reading), then you must simulate the roll of a dice and finally inform the player if he has won or lost (print on screen).

5) Write an algorithm to invert a string of characters.

6) Explain what the following algorithm performs

```
algorithm toInt (String : cad)
      var Number : long,i ← 0,pot ← 1,res ← 0
      var String : c
      long ← length (cad)
      while (i < long) do
            c ← charIn(cad,long - i - 1)
            res ← res + ord(c) * pot
            pot ← pot * 10
            i ← i + 1
      end while
      return res
en algorithm
```

Where **ord** is a function that receives a character and returns its value in numeric format. This way it is compatible to work with arithmetic operations.

7) Propose a modular solution to the problem of determining if a number is prime.

8) Write a sub-algorithm that has a parameter of type number that represents a year and determines whether the year is leap year or not.

9) Write a sub-algorithm to determine the number of days of a month of a year, the latter two parameters of the sub-algorithm.

10) A meteorological station collects rain data for a month and year of a determined locality. An algorithm must be written asking the user to enter a month and a year, and allow for each day of that month in that year, enter the millimetres of water that were recorded (in case of not having registered a rain one day is entered 0) . The amount of total millimetres of water precipitated in that month must be shown, the maximum precipitation recorded for a day and on what day it was given, and finally if it rained two days in a row.