Assignment 2
Part 2: Iterative Structures (Loops)

Exercise 1

Consider the following algorithm:

```
Algorithm Exercise1;
Var A, i, cmp : integer ;
Begin
print ("Give the value of A :") ; read (A) ;
cmp←1 ; i←1 ;
While (i <= A div 2) do
If (A mod i=0) then
cmp←cmp+1 ;
EndIf ;
i←i+1 ;
EndWhile ;
print ("cmp= ", cmp) ;
End.
```

1- Unroll this algorithm and give the display for A=6 then A=9.
2- Indicate what this algorithm does.

Exercise 2

Write the algorithm that determines the minimum of a sequence of n integer numbers ($0 < n < 100$).

Exercise 3

Write an algorithm that allows you to:
1- Display the sum of integer numbers that are between 1 and 100.
2- Display the even integers which are between 1 and 80 (then the odd numbers).
3- Display all natural numbers multiple of 5 that are between 1 and 100 using the 'for' loop and the 'while' loop.
5- Display the sum of n integer numbers ($n>0$) as well as their average.
6- Read an integer $x>0$, then double it as many times until it exceeds 60.
8- Display the minimum between n integer numbers ($n>0$).
9- Calculate the sum of several given integer numbers and stop as soon as the sum exceeds 500, then display their average.

Exercise 4

Write an algorithm that reads n real values between 0 and 20 (which represent the averages of n students with $0<n\leq100$) and displays:
• The percentage of students with an average < 10
• The percentage of students with an average ≥ 10 and < 15
• The percentage of students with an average ≥ 15
The algorithm must take into account incorrect entries (values < 0 and > 20). In this case, the user is asked to re-enter the value until it is correct.

Exercise 5

Write an algorithm that allows you to:

1- Read two positive integers A, B.
2- Calculate their product using only additions.
3- Calculate the quotient and remainder of A by B without using division operators.

Exercise 6

Write an algorithm (or a C program) that determines whether a given integer input from the keyboard is a perfect number. A number is said to be perfect if it is equal to the sum of its divisors, prime or not; we agree to consider 1 as a divisor but not the number itself:

Example:

6 is perfect because 6 = 1 + 2 + 3.
28 is perfect because 28 = 1 + 2 + 4 + 7 + 14.

Exercise 7

Based on the MOD and DIV operators, write an algorithm which:

1- Reads an integer x (x>0).
2- Calculates and displays the number of digits of x and constructs its inverse.

Example: if x=187 then: the number of digits of x is 3, and its inverse is: 781.

Reminder: **187 mod 10 = 7, and 187 div 10 = 18.**