# SW N°3 : REPETITIVES STATEMENTS : LOOPS

**Exercise 01 : Dividers of a number**

- Write an algorithm to read an integer N, greater than 1 and display all its dividers.

**Exercise 02 : Perfect, Deficient or Abundant (Examen 2022/2023)**

A number is perfect if it is equal to the sum of its proper dividers (different from itself). A number is deficient if it is strictly greater than the sum of its proper dividers. A number is abundant if it is strictly less than the sum of its proper dividers.

- Write an algorithm to display whether a given number N is perfect, deficient or abundant.

Example: N = 28 --------> 28 is perfect
N = 16 --------> 16 is deficient
N = 18 --------> 28 is abundant

**Exercise 03 : Maximum**

Write an algorithm that successively asks the user for 20 numbers, and then tells him which was the largest among these 20 numbers.

**Exercise 04 : Division by subtraction**

Write an algorithm allowing you to divide an integer A (positive or zero) by the positive integer B.

Example : we want to divide 20 by 3.

| | | |
|---|---|---|
| 20 – 3 = 17 | 1 | |
| 17 – 3 = 14 | 2 | 2 is the rest. |
| 14 – 3 = 11 | 3 | The quotient is the number of |
| 11 – 3 = 8 | 4 | times we were able to subtract |
| 8 – 3 = 5 | 5 | 3 from 20 (6 times) |
| 5 – 3 = 2 | 6 | |

Dividing A by B involves finding Q and R positive or zero such that: A = B*Q + R with R < B.

**Exercise 05 : Average calculation**

Write an algorithm that asks the user for a series of numbers, representing measurements that can only be strictly positive. The user does not count the number of measurements to enter, but signals that he is finished by entering a negative or a null number.

- The algorithm then displays the number of measurements entered, and their arithmetic average.

**Exercise 06 : Never satisfied (Examen 2014/2015)**

Write an algorithm that asks for a number between 10 and 20, until the answer matches. In the event of a response greater than 20, a message will appear: “Smaller! ”, and conversely, “Bigger!” » if the number is less than 10.

**Exercise 07 : Syracuse sequence (Examen 2014/2015)**

The Syracuse sequence is defined by $u_0 \in \mathbb{N}$ and

$$for\ each\ n \in \mathbb{N}, \quad u_{n+1} = \begin{cases} \frac{u_n}{2}, & if\ u_n\ is\ even \\ 3u_n + 1, & if\ u_n\ is\ odd \end{cases}$$

Write an algorithm asking the user for a number n and displaying all the values $u_1, u_2, \ldots, u_n$ ; with $u_0 = 3$.

**Exercise 08 : The FIBONACCI sequence**

Construct the algorithm which calculates the $N^{th}$ (with N>2) term of the FIBONACCI sequence which is defined by : $\begin{cases} \boldsymbol{U_0 = U_1 = 1} \\ \boldsymbol{U_n = U_{n-1} + U_{n-2}} \quad \text{for every n} \geq 2 \end{cases}$

FIBONACCI sequence: 1, 1, 2, 3, 5, 8, 13, ...
Example : if N=5 then we display : 8

**Exercise 09 : Convert from decimal to binary**

Construct the algorithm for converting an integer N into binary (base 2).

Example : $(29)_{10} = (11101)_2$

**Exercise 10 : Divisibility by 3 (Examen 2020/2021)**

An integer is divisible by 3 if the sum of its digits is divisible by 3.

Write an algorithm that allows you to say whether an integer N is divisible by 3 or not.

Example : N = 903 => Sum of digits = 3+0+9 = 12 is divided by 3; then, 903 can be divided by 3;

**Exercise 11 : the approximation of Sin(x) (Examen 2013/2014)**

Write an algorithm that calculates the approximation of Sin(x), such that :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

With x given real and n positive integer.