

## Imperative is not Iterative

### 1 Conditionals and functions

#### Exercise 1.1 (Maximum)

The following algorithm reads three integer values and displays the maximum of the three.

```
function max3(integer x, y, z) : integer
  variables
    integer    max
begin
  if (x > y) and (x > z) then
    max ← x
  else
    if (y > x) and (y > z) then
      max ← y
    else
      if (z > x) and (z > y) then
        max ← z
      end if
    end if
  end if
  return max
end

variables
  integer    a, b, c

begin      /* main algorithm */
  read (a)
  read (b)
  read (c)
  write (max3 (a, b, c))
end
```

Correct (if necessary) and simplify the algorithm: there must be the fewer tests possible. Then translate it in Python.

---

#### Exercise 1.2 (The day after)

A date is defined by three integers for the year, the month and the day.

Write a script that displays, given a date, the date of the day after. Furthermore, the script has to test whether the given date is valid (ex: there is no February 30<sup>th</sup>).

## 2 Recursive functions

### Exercise 2.1 (List to 9)

Given a 2-digit positive integer  $AB$  such that  $A$  and  $B$  are different.

For example  $AB = 19$ .

- Reverse the 2 digits to obtain 91.
- Subtract 19 from 91 to obtain  $91 - 19 = 72$ .

This process is repeated with 72 (to obtain  $45 = 72 - 27$ ). A last repetition gives  $9 = 54 - 45$ . This list is called "list to 9" because it always ends with the number 9 (unless the two digits were equals, then it returns 0).

Write a Python script that displays the list to 9 of a given number.

---

### Exercise 2.2 (Perfect number)

A perfect number is an integer greater than 1 that is equal to the sum of its divisors excluding the number itself.

Example:  $28 = 1 + 2 + 4 + 7 + 14$ .

Write a script that reads an integer. Then, if the number is valid, it displays the divisors of the number and tests whether it is perfect.

Example of display:

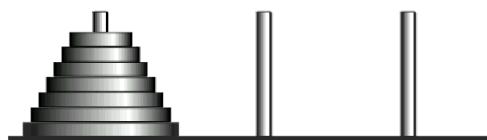
```
1   give n
2   28
3   divisors are: 1, 2, 4, 7, 14,
4   28 is perfect.
```



**Bonus :** write an optimized version that displays the divisors in order.

---

### Exercise 2.3 (Bonus: Tower of Hanoi)



The objective of the puzzle is to move the entire stack from the *source* post to another post (the *destination*), obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack.
- No disk may be placed on top of a smaller disk.

Write a script that solves the problem: it displays the performed moves. The script also displays the number of moves.

How many moves are required with  $n$  disks?