

## Série d'exercices (les structures de contrôle)

### Exercice 1

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

### Exercice 2

1. Ecrire un algorithme en utilisant la boucle **Pour** qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 12, le programme affichera les nombres de 13 à 22.
2. Réécrire l'algorithme précédent, en utilisant cette fois la boucle **Tantque**

### Exercice 3

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

**Table de 7 :**

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

...

$$7 \times 10 = 70$$

### Exercice 4

Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer :

$$1 + 2 + 3 + 4 + 5 = 15$$

NB : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

### Exercice 5

1. Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

**NB :** la factorielle de 8, notée  $8!$ , vaut  $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$

2. Ecrire un autre algorithme qui calcule la puissance  $n$  d'un entier  $x$  ( $x^n$ ).

**NB:**  $5^4 = 5 \times 5 \times 5 \times 5$ .

Ecrire les solutions en utilisant la boucle **Pour**, **Tantque**, et **Repeter**

### Exercice 6

1. Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

etc.

Entrez le nombre numéro 20 : 6

**Le plus grand de ces nombres est : 14**

2. Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre :

**C'était le nombre numéro 2**

3. Réécrire l'algorithme précédent, mais cette fois-ci on ne connaît pas d'avance combien l'utilisateur souhaite saisir de nombres. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

### Exercice 7

1. Ecrire un algorithme qui permet d'afficher **les diviseurs** d'un nombre **entier positif** donné N.
2. Ecrire un algorithme qui permet d'afficher tous **les multiples inférieures à 100** d'un nombre **entier positif** donné N.

### Exercice 8

1. Écrire un algorithme permettant d'afficher les nombres paires compris entre 0 et N, où N saisi par l'utilisateur.
2. Écrire un algorithme qui permet de calculer la somme d'entiers impairs de 1 jusqu'à un entier N saisi par l'utilisateur. **Exemple** N=8 Somme = 1 +3+5+7= 16

### Exercice 9

Le nème terme  $U_n$  de la suite de **Fibonacci** est défini comme la somme des deux termes le précédent c.à.d.

$$U_n = U_{n-1} + U_{n-2}, n \geq 2.$$

Ecrire l'algorithme permettant de calculer le nème terme de la suite de Fibonacci, tel que les termes  $U_0$  et  $U_1$  ainsi que la valeur de n sont donnés par l'utilisateur.