

TD N°3 : Algorithmique et Programmation LE – Informatique et Mathématique

Objectifs : Instructions Itératives : Pour, TantQue et Répéter jusqu'à.

Exercice 1 :

Énoncé :

En justifiant le choix de boucle utilisée, écrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 20, le programme affichera les nombres de 21 à 30.

Justification de choix de la boucle :

L'utilisateur doit fournir un nombre de départ, que l'algorithme doit récupérer et stocker dans une variable entière en utilisant l'instruction **Lire()**. L'affichage des 10 nombres suivants veut dire tout simplement qu'on a une idée à priori des nombres de répétitions, vu qu'on peut définir une valeur initiale et une valeur finale, et dans ce cas l'utilisation de la boucle Pour est possible.

Nous pouvons tout de même utiliser la boucle **TantQue** en définissant une condition qui va nous permettre de rentrer à la boucle et d'en sortir.

Version 1 : En utilisant la boucle Pour

```
Algorithme Affichage
Variables N, i: Entier
Début
    Écrire ("Donnez un nombre entier")
    Lire (N)

    Pour i allant de N+1 à N+10 faire
        Ecrire (i)
    FinPour

Fin
```

Version 2 : En utilisant la boucle TantQue

```
Algorithme Affichage
Variables N, i: Entier
Début
    Écrire ("Donnez un nombre entier")
    Lire (N)
    i ← N+1
    TantQue (i ≤ i+10)
        Ecrire (i)
        i ← i+1
    FinTantQue

Fin
```

Exercice 2 :

Écrire un algorithme qui demande un nombre à l'utilisateur, et qui affiche la table de multiplication de ce nombre. Exemple: Si l'utilisateur entre le nombre 9, le résultat sera sous la forme suivante :

$$\begin{aligned}9 \times 1 &= 9 \\9 \times 2 &= 18 \\&\dots \\9 \times 10 &= 90\end{aligned}$$

```
Algorithme Table_Multiplication
Variables N, i : Entier
Début
    Écrire ("Entrez un nombre")
    Lire (N)
    Écrire ("La table de multiplication de ce nombre est : ")
    Pour i allant de 1 à 10 faire
        Écrire (N, " * ", i, " = ", N*i)
    FinPour
Fin
```

Exercice 3 :

Écrire un algorithme qui calcule et affiche la somme $S = 1 + 2 + 3 + \dots + n$

Le nombre n est un entier à lire au clavier.

Exemple : Si l'utilisateur entre le nombre 6, le résultat calculé est : $1 + 2 + 3 + 4 + 5 + 6 = 21$

```
Algorithme Somme
Variables N, i, Somme : Entier
Début
    Écrire ("Donnez un nombre")
    Lire (N)
    Somme ← 0
    Pour i allant de 1 à N faire
        Somme ← Somme + i
    FinPour
    Écrire ("La somme des nombres de 1 à ", N, "est : ", Somme)
Fin
```

Exercice 4 :

Écrire un algorithme qui demande un nombre entier n au clavier, ensuite calcule et affiche la *factorielle* de n . Exemple : Si l'utilisateur entre le nombre 4, le résultat affiché est 24.

Rappel et explication:

La factorielle de n est $n! = 1 \times 2 \times 3 \times \dots \times n$.

Dans un premier temps, on initialise la factorielle par la valeur 1 qui est $1! = 1$ (**fact** ← 1).

Ensuite pour chaque itération de la boucle *Pour*, on multiplie l'ancienne valeur de **fact** par la valeur courante de i .

```
Algorithme Factorielle
Variables N, i, fact : Entier
Début
    Écrire ("Donnez un nombre entier")
    Lire (N)
    fact ← 1
    Pour i allant de 2 à N faire
        fact ← fact * i
    FinPour
    Ecrire (N, "!=" , fact)
Fin
```

Remarque : La valeur initiale de la boucle Pour est 2 car on a exclu le cas de $i = 1$ en initialisant fact par 1.

Exercice 5 :

Écrire un algorithme qui demande à l'utilisateur un entier n et affiche les *diviseurs* de ce dernier ainsi que la *somme des diviseurs*. Exemple : Si l'utilisateur entre le nombre 10. Ses diviseurs sont : 1, 2, 5, 10. Et la somme des diviseurs est 18.

```
Algorithme Diviseurs
Variable N, i , S: Entier
Début
    Écrire ("Veuillez saisir n : ")
    Lire (N)
    S ← 0
    Pour i allant de 1 à N faire
        Si (N mod i = 0) Alors
            Écrire (i, " est un diviseur de ", N)
            S ← S+i
        FinSi
    FinPour
Fin
```

Exercice 6 :

Écrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 3 *jusqu'à ce que la réponse convienne*.

Version 1 : En utilisant la boucle TantQue

```
Algorithme Affichage_Nombre
Variable n : Entier
Début
    Écrire ("Entrez un nombre entre 1 et 3")
    Lire (n)
    TantQue (n < 1 OU n > 3)
        Écrire ("Saisie erronée. Recommencez")
        Lire (n)
    FinTantQue
    Écrire (n)
Fin
```

Version 2 : En utilisant la boucle Répéter...Jusqu'à

```
Algorithme Affichage_Nombre
Variable n : Entier
Début
    Répéter
        Écrire ("Entrez un nombre entre 1 et 3")
        Lire (n)
    Jusqu'à (n >= 1 ET n <= 3)
        Écrire (n)
Fin
```

Remarque: Nous notons que lorsqu'on utilise la boucle Répéter ...Jusqu'à, on est sûrs que les instructions de la boucle vont être exécutées au moins une seule fois, il s'agit de la première fois, où la condition de la boucle n'est pas encore testée. Contrairement à la boucle TantQue, on ne rentre dans la boucle que si la condition est valide. Ce qui explique la nécessité de demander à l'utilisateur une première valeur de **n** à l'extérieur de la boucle pour la première version utilisant TantQue.

Exercice 7 :

Réécrire l'algorithme de l'exercice précédent, mais cette fois-ci en affichant des messages, si la valeur entrée est inférieure à 1, on affiche "plus grand" et si elle est supérieure à 3 on affiche "plus petit".

```
Algorithme ex7
Variable N : Entier
Début
    N ← 0
    Écrire ("Donnez un nombre entre 1 et 3")
    Lire (N)
    TantQue (N < 1 ou N > 3)
        Si (N < 1) Alors
            Écrire ("Plus grand !")
        Sinon
            Écrire ("Plus petit !")
        FinSi
        Lire (N)
    FinTantQue
Fin
```