

Chapitre 4

I - Boucle tantque :

Algo :

```
Tantque "condition" faire  
    "instruction"  
FinTantque
```

PHP/C :

```
While("condition")  
{  
    "instruction";  
}
```

La boucle tantque exécute un bloc d'instruction autant de fois qu'une instruction reste vérifiée si la condition est au départ une vérifiée, le bloc d'instruction n'est pas exécuté.

Le nombre d'itération ne peut pas être connu à l'avance.

Dans le corps de la boucle , il faut prévoir une instruction d'exécution de la condition sinon on aura une boucle infinie.

Exemple : Ecrire un algo qui affiche les nombres paires compris entre 2 et 20.

Algo : nbpair

Declaration :

 |nb : entier

début :

 |nb ← 2

 |tantque nb<=20 faire

 |afficher(nb)

 |nb ← nb+2

 |fintantque

fin nbpair

II - La boucle fair-tantque :

Algo :

```
faire
    "instruction"
tantque "condition"
```

PHP/C

```
do {
    "instruction";
} while("condition")
```

La boucle **faitantque** est un schéma inverse de la boucle **tantque**.

Le test de la condition est postérieur à l'exécution de l'instruction. De ce fait, l'instruction est au moins exécutée une fois, voir plusieurs fois.

Exemple : Écrire en algo qui affiche nombres pair compris entre 2 et 20.

```
Algo: nbpair
Déclaration:
    nb : pair
Début :
    nb ← 2
    faire : Afficher (nb)
        nb ← nb + 2
    tantque nb <= 20
Fin nbpair
```

III - La boucle Pour

Algo :

```
Pour : "indice allant de" ValeurDébut à ValeurFin Pasde Pas faire
    "instruction"
FinPour
```

PHP/C :

```
for("initialisation"; "condition"; "instruction")
{
    "instruction";
}
```

La boucle **Pour** exécute un bloc d'instruction un nombre de fois connu à l'avance, calculé en utilisant la ValeurDébut, la ValeurFin et le Pas.

Par défaut le Pas est égal à 1 ou -1.

Exemple : Écrire un algo qui affiche les nombres pairs compris entre 2 et 20.

```
Algo : nbpair
Déclaration :
    nb : entier
```

Début :

```
Pour nb allant de 2 à 20 pas de 2 faire
Afficher(nb)
FinPour
```

Fin nbpair

Exercice 1:

Ecrire un algo, un prog C, PHP, qui permet de savoir un nombre et une limite et afficher la table de multiplication du nombre jusqu'à la limite.

Algo : exo 1

Declaration :

```
| nbr, limite, res, m : entier
```

Début :

```
|Afficher("Saisir votre chiffre à multiplier")
|Saisir(nbr)
|Afficher("Saisir votre limite de multiplication")
|Saisir(limite)
|Pour m allant de 1 à limite faire
|res=nbr*m
|Afficher(m,".",nbr,"=",res)
|FinPour
```

Fin exo1

Exercice 2:

Ecrire un algo, un prog c, un PHP ui saisi un nombre et affiche ses divisions.

Algo : exo2

Declaration :

```
| nbr, res, d : entier
```

Début :

```
|Afficher("Saisir votre chiffre à diviser")
|Saisir(nbr)
|Pour d allant de 1 à nbr limite faire
|    Si nbr%d=0 alors
|        Afficher("le diviseur est : "d)
|    FinSi
|FinPour
```

Fin exo2