

SÉANCE 2 : STRUCTURES RÉPÉTITIVES

Exercice 1 : Boucles while (tant que), boucle for (pour)

1. Que réalise le premier programme Python ? Fournir le déroulé par ligne et l'affichage réalisé.
2. Que réalise le deuxième programme Python ? Fournir le déroulé par ligne et l'affichage réalisé.
3. Comment écrire l'intervalle des valeurs affichées ?

```
1  # VAR (déclaration)
2  depart:int;
3  fin_exclue:int;
4  k:int;
5  # début
6  depart      = 5 ;
7  fin_exclue = 10 ;
8  k = depart ;  # k valeur courante
9  print("Départ") ;
10 while (k < fin_exclue):
11     print(k) ;
12     k = k + 1 ;
13 print("Fin") ;
14 # fin
```

```
1  # VAR (déclaration)
2  depart:int;
3  fin_exclue:int;
4  # début
5  depart      = 5 ;
6  fin_exclue = 10 ;
7  print("Départ") ;
8  for k in range(depart, fin_exclue):
9      print(k) ;
10 print("Fin") ;
11 # fin
```

Exercice 2 : Boucle et alternative

1. Que réalise ce programme ?
2. Modifiez le en remplaçant le while par un for.
3. Modifiez le pour qu'il affiche les nombres qui sont à la fois multiples de 3 ou de 7 dans l'intervalle [0, 100].
4. Modifiez le pour qu'il affiche les nombres qui sont multiples de 3 mais pas de 7.

```
1  # VAR (déclaration)
2  depart:int;
3  fin_exclue:int;
4  k:int;
5  # début
6  depart_inclus = 5 ;
7  fin_exclue    = 10 ;
8  k = depart_inclus ;
9  print("Départ") ;
10 while (k < fin_exclue):
11     if (k % 2 == 0):
12         print(k) ;
13     k = k + 1 ;
14 print("Fin") ;
15 # fin
```

Exercice 3 : Informagie, le retour

Créer un programme qui affiche tous les nombres de 0 à 31 nécessitant que le bit du 4 soit activé en notation binaire.

Exercice 4 : Nombres premiers

1. Qu'est-ce qu'un nombre premier ?
2. Réaliser un programme qui demande un nombre n puis tente de diviser ce nombre par tout entier situé entre 2 et $(n-1)$. Si n est divisible par l'un de ces entiers, afficher l'entier diviseur de n .
3. Réaliser un programme qui signale si un entier lu au clavier est premier ou pas. Pour cela, il faudra :
 - initialiser une variable booléenne `premier` à Vrai (True en Python) pour signaler qu'on considère au début que le nombre est premier.
 - tenter de diviser n par tout entier entre 2 et $(n-1)$ et mettre la variable `premier` à Faux (False en Python).
 - en fin de programme, on affichera un message permettant de savoir si n est premier ou pas.
4. Si ce n'est pas encore fait, optimiser le programme pour qu'il stoppe le parcours dès qu'on détecte que le nombre n'est pas premier.

Exercice 5 : Approximation de $\sqrt{2}$

On considère la suite (u_n) définie par $u_0 = 1$ et pour tout entier naturel n , $u_{n+1} = \frac{1}{2} \left(u_n + \frac{2}{u_n} \right)$.

Cette suite converge vers $\sqrt{2}$.

1. Écrire un programme Python qui :

- initialise u à 1 ;
- demande à l'utilisateur une précision `epsilon` ;
- calcule successivement les termes de la suite ;
- s'arrête lorsque l'écart entre u et $\sqrt{2}$ est inférieur ou égal à `epsilon` ;
- affiche le nombre d'itérations effectuées ainsi que la valeur obtenue.

Exemple d'affichage :

Précision : 0.000001
Nombre d'itérations : 4
Valeur obtenue : 1.414213562373095

On pourra utiliser la fonction `sqrt` du module `math` :

```
1  from math import sqrt
2
3  x:float ;
4  x = sqrt(2) ;
5  print(x) ;
```

Exercice 6 : Preuve de terminaison [NSI]

Démontrer que l'algorithme de l'exercice 1 s'arrête quelles que soient les valeurs `debut` et `fin_exclue`.