



Matière : Informatique 2 (S2)

TP N°1: Structures répétitives: boucle for (2)

Semaine 2 (SM5, SM6, SM7 et SM8)

Rappel:

for **compteur** in range (valeur de départ, valeur d'arrêt, pas):

```

instruction 1
instruction 2
.
.
instruction N
    
```

Remarques:

- compteur** est une variable qui va recevoir les valeurs de **valeur de départ** jusqu'à **valeur d'arrêt** selon le pas qui lui est donné. La première valeur donnée au compteur est **valeur de départ** ensuite **compteur=compteur+pas** jusqu'à ce que **compteur** arrive à la **valeur d'arrêt**.
A chaque fois que le compteur reçoit une valeur, les instructions qui suivent la boucle à savoir: **instruction1, instruction2, ...instruction N** sont exécutées.
- Si on utilise un **pas positif** :
valeur de départ < valeur d'arrêt
La dernière valeur que va recevoir le compteur est **valeur d'arrêt-1 (si vous voulez aller jusqu'à 10 il faut mettre 11 comme valeur d'arrêt)**
Exemple: **for i in range (0,11,2):** i reçoit 0 ensuite 2 ensuite 4 ensuite 6 ensuite 8 ensuite 10 car le pas est 2 et la valeur d'arrêt est: 11-1 et donc 10.
- Si on utilise un **pas négatif**:
valeur de départ > valeur d'arrêt.
la dernière valeur que va recevoir le compteur est valeur d'arrêt+1 (si vous voulez aller de 10 jusqu'à 0 il faut mettre -1 comme valeur d'arrêt).
Exemple: **for i in range (10,-1,-2):**i reçoit 10 ensuite 8 ensuite 6 ensuite 4 ensuite 2 ensuite 0 car le pas est -2 et la **valeur d'arrêt** est -1+1 et donc 0
- Si on ne précise pas le pas par exemple **for i in range (0,10):** le pas est considéré égale à 1.
- Si on ne précise pas la valeur de départ par exemple **for i in range (10):** la première valeur que prendra le compteur est 0 et le pas est considéré égale à 1.

Exercice 1: Exécutez chaque boucle for présentée dans le tableau ci-dessous et notez la valeur de départ, la valeur d'arrêt et le pas correspondant.

Boucle	Valeur de départ	Valeur d'arrêt	Pas
for i in range(5,20): print(i)			
for i in range(8): print(i)			
for i in range(-5,11,2): print(i)			
for i in range(15,0,-4): print(i)			

Exercice 2: Testez chaque boucle for dans le tableau qui suit déduisez la valeur de départ et la valeur d'arrêt. Si le pas est positif déduisez la boucle correspondante avec un pas négatif et réciproquement.

Boucle avec un pas positif	V D	V A	Boucle avec un pas négatif	VD	VA
for i in range(0,15): print(i)					
			for i in range(30,-1,-1): print(i)		
for i in range(-100,101,2): print(i)					
			for i in range(-20,-31,-3): print(i)		

Exercice 3:

Ecrire un programme qui demande à l'utilisateur de saisir un entier N, le programme doit afficher ensuite:

- 1) Si N est **positif** et **pair**: les *entiers positifs pairs inférieurs* ou égale à N inclus dans [0,N], dans l'ordre croissant.
Le programme doit afficher la somme **S1** des entiers positifs pairs inclus dans [0,N].
- 2) Sinon si N est **négatif** et **impair**: les *entiers négatifs impairs supérieur* ou égale à N inclus dans [N,-1], dans l'ordre décroissant.
Le programme doit afficher la somme **S2** des entiers négatifs inclus dans [N,-1].

Par exemple:

- ✓ si l'utilisateur saisit N=10 le programme doit afficher les valeurs: 0 2 4 6 8 10 ainsi que la somme des valeurs S1= 30
- ✓ si l'utilisateur saisit N=-9 le programme doit afficher les valeurs: -1 -3 -5 -7 -9 ainsi que la somme des valeurs S2=-25

Exercice 4:

Ecrire un algorithme permettant d'afficher les nombres parfaits compris entre 1 et N.

Un nombre parfait est un entier naturel non nul qui est égal à la somme de ses diviseurs stricts (le nombre lui-même ne fait pas partie des diviseurs).

Exemple : 28 est un nombre parfait puisque 1, 2, 4, 7 et 14 sont les diviseurs stricts de 28 et $14 + 7 + 4 + 2 + 1 = 28$.

Exemple:

Si l'utilisateur saisit 1000 pour N, le programme devra afficher les nombres parfaits compris entre 1 et 1000 à savoir: 6, 28 et 496.

Ci-dessous un exemple d'exécution du programme:

```
===== RESTART: C:\Python 2019-2020\EXOS PYTHON\Exercice4.py =====
Entrez un entier positif 100
6 est un nombre parfait
28 est un nombre parfait
>>> |
```