

Session d'exercices – Les boucles

1. [Difficulté: **] Écrivez un programme qui lit une valeur entière n , $n \geq 0$, calcule la somme des n premiers nombres impairs positifs, et l'affiche à l'écran.

Par exemple, si $n = 5$, le résultat sera $1 + 3 + 5 + 7 + 9 = 25$ et votre programme affichera

Sum of the first 5 odd numbers = 25.

Pour tous les $n < 0$, comme résultat le programme doit afficher seulement

Invalid input value!

et s'arrêter.

Essayez d'implémenter ce programme de 2 façons différentes:

Option a) Implémentez ce programme en utilisant une boucle `while`

Option b) Implémentez ce programme en utilisant `range()` et une boucle `for`

Modifiez le programme pour que chaque deuxième nombre impair soit déduit au lieu d'être ajouté. Par exemple, si $n = 5$, le résultat devient $1 - 3 + 5 - 7 + 9 = 5$.

Indice: Afin de quitter le programme avant le fin, il faut d'abord importer le module appelé `sys`, c.-à.-d. ajouter la ligne `import sys` au début du programme. Ensuite, il faut appeler la méthode `exit()` (`sys.exit()`) pour quitter le programme.

2. [Difficulté: *] Écrivez un programme qui lit une valeur entière n , $n \neq 0$, et affiche en ordre décroissant les premiers n nombres impairs positifs ou négatifs, selon le signe de n . Par exemple, si $n = 5$, le programme doit afficher:

9 7 5 3 1

Si $n = -5$, le programme doit afficher:

-1 -3 -5 -7 -9

Si $n = 0$, le programme ne doit rien afficher.

Indice: Par défaut, la fonction `print` affiche le string passé au paramètre à la ligne (c.-à.-d. qu'elle ajoute le caractère de fin de ligne à la fin du string). Cependant, il est possible de spécifier un autre caractère comme la fin. Par exemple, on peut terminer l'affichage du string avec une virgule comme suit : `print("mon string", end = ",")`

3. [Difficulté: **] Écrivez un programme qui lit deux valeurs entières n et m ($n \geq 1$ et $m \geq 1$) et affiche la multiplication équivalente et le résultat de n^m (n à la puissance m). Par exemple, si $n = 3$ et $m = 4$, le programme doit afficher

3 * 3 * 3 * 3 = 81

.....

Si soit $n < 0$, soit $m < 0$, le programme ne doit rien afficher.

4. [Difficulté: *] Écrivez un programme pour lire des nombres entiers tant que le nombre donné n'est pas négatif. Le programme doit afficher la valeur de nombre dernier, c.à.d. la seule valeur négative entrée. Par exemple, si les nombres suivants sont entrés:

1
99
5
-3

le programme doit afficher seulement -3.