

07. Boucle *pour*

Une boucle *pour* permet de répéter une instruction un nombre de fois déterminé, elle dispose d'un *compteur* d'itérations.

Exemple 1. Calcul de 2^n .

Entrée: variable entière n
Sortie: variable entière p dont la valeur est égale à 2^n
Début
$p \leftarrow 1$
Pour k allant de 1 à n faire
$p \leftarrow 2p$
FinPour
Fin

	valeur de p
Début	1
$k = 1$	2^1
$k = 2$	2^2
\vdots	
$k = n$	2^n
Fin	2^n

Le compteur d'itérations peut être utilisé dans l'instruction itérée.

Exemple 2. Calcul de $1 + 2 + \dots + n$.

Entrée: variable entière non nulle n
Sortie: variable entière s dont la valeur est égale à $1 + 2 + \dots + n$
Début
$s \leftarrow 0$
Pour k allant de 1 à n faire
$s \leftarrow s + k$
FinPour
Fin

	valeur de s
Début	0
$k = 1$	1
$k = 2$	$1 + 2$
\vdots	
$k = n$	$1 + 2 + \dots + n$
Fin	$1 + 2 + \dots + n$

Exercice 1. Écrire un algorithme permettant d'afficher les entiers de 1 à n dans l'ordre décroissant.

Exercice 2. Écrire un algorithme permettant de calculer $n!$ pour $n \in \mathbb{N}^*$.

Exercice 3. Écrire un algorithme permettant d'afficher les n premiers entiers impairs.

Exercice 4. Écrire un algorithme permettant de calculer le n -ième terme de la suite définie par

$$\begin{cases} u_1 = 1 \\ u_{n+1} = u_n + \frac{1}{u_n}, n \in \mathbb{N}^* \end{cases}$$

Exercice 5. Écrire un algorithme permettant de calculer la somme des n premiers entiers impairs.

Exercice 6. Écrire un algorithme permettant de calculer le n -ième terme de la suite de Fibonacci définie

$$\text{par } \begin{cases} u_1 = 1 \\ u_2 = 1 \\ u_{n+2} = u_{n+1} + u_n, n \in \mathbb{N}^* \end{cases}$$

Exercice 7. Écrire un algorithme permettant de déterminer si un entier $n \geq 2$ est un nombre premier.