

LA RÉCURSIVITÉ

Citation : « Le génie est fait de dix pour cent d'inspiration et de quatre-vingt-dix pour cent de transpiration » Thomas Edison :

- Fonction qui s'appelle pour résoudre un problème en le décomposant en versions plus petites et plus simples du même problème. La programmation récursive est une **technique de programmation qui remplace les instructions de boucle (while, for, etc.) par des appels de fonction.**
- La récursivité est une manière simple et élégante de résoudre certains problèmes algorithmiques, notamment en mathématique, mais cela ne s'improvise pas, il convient donc de savoir comment ce principe fonctionne.
- L'approche récursive est un des concepts de base en informatique.
- On oppose généralement les algorithmes récursifs aux algorithmes itératifs, qui eux, utilisent plutôt des boucles pour et des boucles tant que, pour répéter des opérations.
- Une récursivité mal formulée équivaldrait à une boucle infinie.
- Les deux points majeurs :
 - Au moins un cas de base/condition d'arrêt/règle de sortie : pour arrêter les appels récursifs ;
 - Au moins un cas général/cas récursif : Une étape récursive qui réduit la taille du problème jusqu'à ce que le cas de base soit atteint.
- Exemple : calcul de la factorielle n ($n!$)

Boucle for (merci d'écrire les formes while et do...while)	
<pre>int factorielle(int n) { int F, i ; if (n <=1){ F = 1 ; }else{ F = 1 ; for(i=2 ; i<=n ; i++){ F = F * i ; } } return F ; }</pre>	
	Cas initial
	Cas itératif
Solution récursive	
<pre>long factorielle(int n) { long z; if (n == 0){ z = 1; }else{ z = n * factorielle(n-1); } return z; }</pre>	
	Cas de base Règle de sortie
	Cas récursif Cas général

- Exercice 1

Écrire un programme qui permet de déterminer, au moyen d'une fonction, le nombre de combinaisons possibles de p éléments pris parmi n , sachant que ce nombre peut être calculé par la formule 1 :

$$C_n^p = \frac{n!}{p! (n - p)!}$$

et par la formule 2 :

$$\left[\begin{array}{l} C_n^p = C_{n-1}^{p-1} + C_{n-1}^p \\ C_n^n = 1 \\ C_n^1 = n \end{array} \right] C_n^p \left\{ \begin{array}{ll} 1 & \text{Si } p = n \\ n & \text{Si } p = 1 \\ C_{n-1}^{p-1} + C_{n-1}^p & \text{Sinon} \end{array} \right.$$

- Exercice 2

Écrire un programme qui contient les deux procédures récursives suivantes :

- Procédure **montee** qui affiche les entiers de 1 à n , dans l'ordre croissant ;
- Procédure **descente** qui affiche les entiers de 1 à n , dans l'ordre décroissant.