

Chapitre 9 : Récursivité

Contenu du chapitre :

1. Définition de la récursivité.
2. Forme générale d'un sous algorithme récursif.
3. Du particulier au general.

1. Définition

On dit qu'un sous algorithme est récursif lorsqu'il s'appelle lui-même de façon répétitive jusqu'à ce qu'une condition donnée soit vérifiée.

Les sous algorithmes récursifs sont utilisés lorsqu'on effectue des calculs de types récursifs.

2. Forme générale d'un sous algorithme récursif

Un sous algorithme récursif possède la forme d'un sous algorithme usuel. Cependant, il est caractérisé par un appel à lui-même. De plus, son corps englobe *un cas général* et *un (ou plusieurs) cas particulier(s)*.

Exemple explicatif : Soit la fonction factorielle, dont la notation mathématique est définie comme suit : $n! = 1 \times 2 \times \dots \times n$, sachant que $0! = 1! = 1$.

Pour avoir le résultat de $n!$ il faut calculer $(n-1)!$ puis le multiplier par n . Et pour calculer $(n-1)!$ il faut calculer $(n-2)!$ puis le multiplier par $(n-1)$. Et ainsi de suite.

D'une façon générale : $n! = n \times (n-1)!$ (calcul récursif).

Le sous algorithme contenant une telle fonction s'écrira donc comme suit :

```
Algorithme Récursivité ;  
Var n, r ; Entier ;  
Fonction Factorielle (n : Entier) : Entier ;  
Début  
    Si (n=0) ou (n=1) alors  
        Factorielle  $\leftarrow$  1;  
    Sinon  
        Factorielle  $\leftarrow$  n * factorielle (n-1);  
    Fin Si ;  
Fin ;  
Début  
    Ecrire ('Entrez un nombre') ;  
    Lire (n) ;  
    r  $\leftarrow$  Factorielle (n); {Appel de la fonction factorielle}  
    Ecrire (n, ' ! = ', r) ;  
Fin
```

La fonction **Factorielle (n)** va s'appeler elle-même avec la valeur **(n-1)**. La fonction **Factorielle (n-1)** va à son tour, s'appeler elle-même avec la valeur **(n-2)**. Et ainsi de suite jusqu'à **n** égale à **1**. Dans ce cas, **Factorielle (1)** vaut **1**, et l'appel récursif s'interrompt. Le compilateur va effectuer donc tous les autres calculs laissés en suspend.

Voici le déroulement du calcul récursif de 4 !

Factorielle (4) $\rightarrow 4 \times$ Factorielle (3)
 $\rightarrow 4 \times 3 \times$ Factorielle (2)
 $\rightarrow 4 \times 3 \times 2 \times$ Factorielle (1)
 $\rightarrow 4 \times 3 \times 2 \times 1$
 $\rightarrow 24$

3. Du particulier au général

Si on observe bien l'exemple explicatif précédent, on déduit que comme dans le cas d'une boucle, le sous algorithme récursif doit comprendre :

- Au moins un cas d'arrêt où l'on ne fait pas d'appel récursif. Ce cas est appelé : *cas particulier du sous algorithme récursif*.
- Au moins un cas dans lequel le sous algorithme s'appelle lui-même pour refaire le traitement. Ce cas est appelé : *cas général du sous algorithme récursif*.

Syntaxe de déclaration :

Procédure *ProcRec* (*paramètres*)

Début

Si (Cas particulier) **alors**

Instructions du point d'arrêt ;

Sinon

{Appel récursif instructions}

Instruction(s) contenant *ProcRec* (*paramètres changés*) ;

Fin Si ;

Fin ;