

## Chapitre V : La récursivité

### V.1. Définition

La récursivité est la possibilité pour un programme de s'appeler lui-même.

Cette notion fondamentale est souvent mal exploitée car mal comprise. On notera que la plupart de débutants en informatique ont une phobie inexplicable vis-à-vis de la récursivité.

La définition mathématique suivante est pourtant facilement assimilée :

$$N! = \begin{cases} 1 & \text{si } N=0 \\ N*(N-1)! & \text{si } N>0 \end{cases}$$

- C'est la définition par *récurrence* du factoriel : on utilise la fonction factorielle pour définir la fonction factorielle elle-même.

Prenons une autre fonction :

$$A^n = \begin{cases} 1 & \text{si } n=0 \\ A * A^{n-1} & \text{si } n>0 \end{cases}$$

- C'est la définition par *récurrence* de la fonction puissance: on utilise la fonction puissance pour définir la fonction puissance elle-même.
- Pour chaque sous programme récursif, on définit deux choses :
  1. La condition d'arrêt (n=0)
  2. La condition de récursivité (n>0)

### V.2. Algorithmes

#### V.2.1. Le factoriel

##### a. Algorithme itératif

```

fonction fact ( n :entier) : entier ; var i :entier ;
  debut
  | fact ← 1 ;
  | pour i allant de 1 jusqu'à n faire
  | fact ← fact*i ;
  | finpour ;
  fin;
  
```

### b. Algorithme récursif

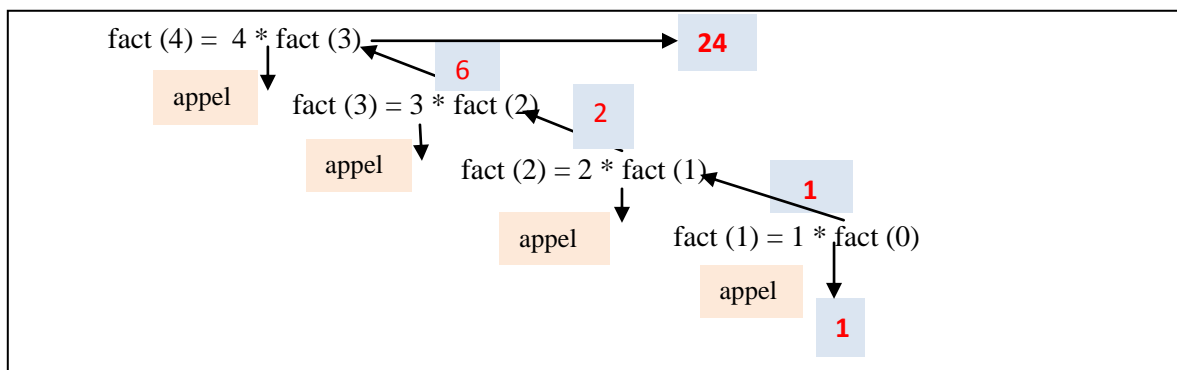
```

fonction fact ( n :entier ) : entier ;
debut
| Si ( n = 0 ) alors
| | fact ← 1
| sinon
| | fact ← n * fact(n-1) ;
| finsi ;
fin;

```

### c. Schéma d'exécution

Soit à calculer fact(4) par la fonction récursive :



## V.3. Fonctionnement de la récursivité

- La récursivité suppose qu'il y a plusieurs appels en suspension (càd pas encore terminés) au même programme.
- Comment le système se souvient-il de l'adresse de retour et des valeurs des variables locales et des paramètres ? Il utilise pour cela une pile appelée **pile des appels**.
- Il faut noter que cette pile est utilisée pour gérer les appels récursifs ou non récursifs.
- A chaque appel, le système **empile** les informations suivantes appelées '**bloc d'activation**' :
  1. Les paramètres
  2. Les variables locales
  3. L'adresse de retour
- A chaque retour, le système **dépille** un bloc d'activation.

- Quand la pile des appels est *vide*, il n' y a plus d'appels en suspension.

### V.3.1. Gestion de la pile des appels de la fonction factorielle

<b>Appel</b> fact(4)	<b>empiler</b> (n=4 )
<b>Appel</b> fact(3)	<b>empiler</b> (n=3)
<b>Appel</b> fact(2)	<b>empiler</b> (n=2)
<b>Appel</b> fact(1)	<b>empiler</b> (n=1)
<b>Appel</b> fact(0)	<b>empiler</b> (n=0)
<b>Retour</b> fact=1	<b>dépiler</b>
<b>Retour</b> fact=1	<b>dépiler</b>
<b>Retour</b> fact=2	<b>dépiler</b>
<b>Retour</b> fact=6	<b>dépiler</b>
<b>Retour</b> fact= 24	<b>dépiler</b>
<b>Pile_vide</b> = vrai	