

La récursivité

Définition :

Une fonction ou une procédure est dite récursive si elle forme avec elle-même une boucle d'appels.
Pour construire une récursivité il faut déterminer :

- Une relation récurante entre l'étape N et l'étape N-1
- un critère d'arrêt des appels

Exemple

Déroulement sur un exemple : le factoriel

Écrire une fonction récursive qui calcule le factoriel d'un nombre $n : n! = 1*2*3*...*n$

Écrivons la fonction itérative En utilisant une boucle :

```
fonction factoriel_it(n:entier) : entier
Déclaration
  | f, i : entier
Début
  | f <- 1
  | pour I allant de 1 à n faire
  | | f <- f * I
  | fin pour
  | retourner f
fin factoriel_it
```

Construisons la récursivité :

- Relation récurrente :
 $n! = 11*2*3*...*(n-2)*(n-1)*n$
 $5! = 1*2*3*4*5$
 $n! = (n-1)! * n$ (relation récurrente)
 $5! = 4!*5$

Pour calculer le factoriel d'un nombre, on calcul le factoriel du nombre précédent qu'on multiplie par le nombre :

- critère d'arrêt

$5! = 4!*5$
 $4! = 3!*4$
 $3! = 2!*3$
 $2! = 1!*2$ $1! = 1$ (critère d'arrêt)

```
fonction factoriel_recu (n : entier) : entier
Déclaration
  | f : entier
Début
  | Si n = 1
  | | alors f <- 1
  | | sinon f <- factoriel_recu(n-1) * n
  | fin si
  | retourner f
fin factoriel_recu
```

```
int factoriel_recu(int n) {
  int f;
  if (n == 1) f = 1;
  else f = factoriel_recu(n-1) * n;
  return f;
}
```

On peut éliminer la variable f en faisant un retour direct

```
int factoriel_recu(int n) {
  if (n == 1) return 1;
  else return factoriel_recu(n-1) * n;
}
```

```
}
```

On peut utiliser l'opérateur ternaire ?

```
int factoriel_recu (int n) {
    return (n == 1)?1:factoriel_recu(n-1)*n;
}
```

Exercices

Exo 1 :

ecire une fonction récursive qui calcul la puissance de a par b par multiplication successives;

```
fonction factoriel_puissance(a, b: entier) : entier
Déclaration
    | p : entier
Début
    | si b = 0
    | | alors p <- 1
    | | sinon p <- factoriel_puissance(a, b-1)*a
    | fin si
    | retourner p
fin factoriel_puissance
```

```
#include<stdio.h>
int factoriel_puissance(int a, int b) {
    return(b == 1)?1:factoriel_puissance(a, b-1) * a;
}
int main() {
    int x, y;
    printf("Donner un nombre: ");
    scanf("%d", &x);
    printf("Donner un exposant : ");
    scanf("%d", &y);
    printf("Le résultat est : %d", puissance(x, y));
    return 0;
}
```

Exo 2 :

Ecrire la fonction récursive qui calcule le nombre de fibonacci avec les données suivantes :
 $f(0) = 1$ et $f(1) = 1$ $f(n) = f(n-1) + f(n-2)$

```
fonction fibo(a : entier) : entier
Déclaration
    | f : entier
Début
    | si a = 0 ou a = 1
    | | alors f <- 1
    | sinon
    | | f <- fibo(a-1)+fibo(a-2)
    | fin si
    | return f
fin fibonacci
```

```
int fibo (int n) {
    return (n <= 1)?1:fibo(n-1)+fib(n-2);
}
```