

### Exemple de fonction recursive :

```
let rec pgcd (a:int) (b:int) : int =  
if b = 0 then a  
else (pgcd b (a mod b));;
```

*inutile en dehors du top level.*

- Une fonction rec est terminale si la dernière opération est l'appel de la fonction elle-même avec ses propres paramètres et pas de variable externe

*je ne suis pas sûr de comprendre ce que vous voulez dire par là*

### Fonction recursive terminale de factorielle :

```
let rec fact_term (n:int) (acc: int) : int =  
if n = 0 then acc  
else (fact_term (n-1) (n*acc));;
```

### Fonction terminale avec recursivité terminale locale de factorielle :

```
let fact(n:int):int =  
let rec fact_term (n:int)(acc:int) :  
  int =  
  if n = 0 then acc  
  else (fact_term (n-1) (n*acc))  
in fact_term n 1
```

*on dirait plutôt "factorielle avec fonction locale récursive terminale"*

let in est utilisé pour rendre une fonction locale

•

Pour rendre une fonction terminale

:

1. On définit une fonction globale non-recursive : let fact
2. On définit une fonction locale terminale de type **let .. in** : let rec fact\_term
3. On rappelle notre fonction locale après le **in** avec le paramètre de la fonction globale : le **n** de let fact

*pour toute déclaration locale, pas uniquement les fonctions locales*

### Fonction recursive pour somme :

```
let rec sum_n (n:int) : int =  
if n=0 then 0  
else n+(sum_n (n-1));;
```

### Fonction recursive terminale de somme :

```
let rec sum_n_term(n:int)  
(a:int):int=
```

```
if n=0 then acc
else (sum_n_term (n-1)(n+acc));;
```

attention - ici vous avez appelé  
votre paramètre a et  
non acc.

### Fonction avec recursivité locale de somme :

```
let sum_n(n:int) :int =
let rec sum_n (n:int): int =
  if n=0 then 0
  else (n + sum_n(n-1)) in
  if n < 0 then raise (Invalid_argument "sum_n")
  else (sum_n n);;
```

### Fonction avec recursivité terminale locale de somme :

```
let sum_n (n:int):int =
let rec sum_n_term (n:int) (a:int):int=
  if n=0 then a
  else (sum_n_term (n-1) (n+a))
in if n<0 then raise (Invalid_argument "sum_n")
  else
    (sum_n_term n 0);;
```

L'exception **raise** abort le program à l'endroit de l'exception