

## TD2:

### Exercice 1:

1.  $\text{let}_{\text{rec}} \text{fact}(n:\text{int}) : \text{int} =$   
if  $(n = 0)$  then 1  
else  $n * (\text{fact}(n-1))$

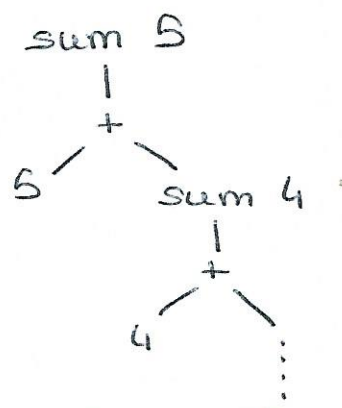
let fact n =  
let rec loop n r =  
if  $(n = 0)$  then r  
else loop  $(n-1) (n * r)$   
in loop n 1

$\Rightarrow$  méthode d'accumulation

Fonction récursive: obligation d'écrire "rec"

### Exercice 2:

1.  $\text{let}_{\text{rec}} \text{sum}_n(n:\text{int}) : \text{int} =$   
if  $(n = 0)$  then 0  
else  $n + (\text{sum}(n-1))$



2.  $\text{let rec sum}_n(n:\text{int}) : \text{int} =$   
if  $(n < 0)$  then raise (Invalid\_argument "sum\_n")  
↓  
permet d'arrêter la fonction.

### Autre façon d'écrire:

let sum\_n\_1  $(n:\text{int}) : \text{int} =$   
let rec aux  $(n:\text{int}) : \text{int} =$   
if  $(n = 0)$  then 0  
else  $n + (\text{aux}(n-1))$   
in if  $(n < 0)$  then raise ( . . . )  
else aux n

3.  $\text{let rec sum}_p(n:\text{int}) : \text{int} =$   
(\* hypothèse:  $n \geq 0$  \*)  
if  $(n = 0)$  then 0  
else  $n * 2 + (\text{sum}_p(n-1))$