

TD 4

Correction et terminaison

Exercice 1 :

1. Écrire la fonction récursive `fact`, calculant la valeur de factoriel de n placé en argument.

```
fact : int -> int
```

2. Prouver la terminaison de la fonction.

Exercice 2 :

On considère la fonction suivante :

```
let rec binome n p = match (n, p) with
| n, 0 -> 1
| n, p -> if (p>n) then 0 else binome (n-1) p + binome (n-1) (p-1);;
```

Prouver la terminaison de la fonction.

Exercice 3 :

On considère la fonction suivante :

```
let rec ackermann n p = match (n, p) with
| 0, p -> p + 1
| n, 0 -> ackermann (n - 1) 1
| n, p -> ackermann (n - 1) (ackermann n (p - 1));;
```

Prouver la terminaison de la fonction.

Exercice 4 :

On appelle inversion paire d'une liste d'entier non vide de longueur paire deux éléments successifs tels que :

- Le premier d'entre eux est situé en position paire (en partant de la position 0).
- Le premier est strictement supérieur au second.

Par exemple :

```
let l_1 = [ 3;7;1;5;9;2;5;4];;
let l_2 = [ 3;7;1;5;4;8;1;3];;
```

`l_1` a une inversion paire, mais pas `l_2`

1. Écrire la fonction `a_une_inversion` qui teste la présence d'une inversion paire d'une liste d'entier non vide de longueur paire.
2. Prouver la fonction.