

Exercice 1. Soit la fonction suivante :

```
let XX = function (0,_) -> 0
                  | (_,0) -> 5
                  | (1,x) -> x+50
                  | (x,1) -> x+35
                  | _     -> 3 ;;
```

Que calculent : $XX(0,1)$, $XX(0,0)$, $XX(1,0)$, $XX(1,4)$, $XX(4,1)$, $XX(25,5)$, $(XX\ 0\ 0)$?

Exercice 2. Définir, en utilisant le filtrage les fonctions fibonacci et puissance.

Exercice 3. Définir, en utilisant le filtrage de 3 manières différentes, la fonction suivante :

$$(n,m) \rightarrow \begin{cases} 1 & \text{si } n > m \\ 0 & \text{si } n = m \\ -1 & \text{si } n < m \end{cases}$$

Exercice 4. Ecrire et typer les fonctions suivantes :

1. `somme liste`: calcule la somme des éléments d'une liste d'entiers;
2. `nb_occur liste elem`: calcule le nombre d'occurrences d'un d'élément dans une liste;
3. `appartient elem liste`: teste l'appartenance d'un élément à une liste;
4. `ensemble liste`: calcule l'ensemble des éléments contenus dans une liste (le résultat est une liste contenant une seule occurrence de chaque élément de la liste `liste`);
5. `regroupe liste`: calcule la liste des couples formés, pour chaque élément distinct de la liste, de cet élément et de son nombre d'occurrences;
6. `taille liste`: calcule la taille de la liste en utilisant `regroupe`, `somme` et `map`;
7. `inclus liste1 liste2`: teste l'inclusion d'une liste dans une autre. Attention aux occurrences multiples: la liste `[1; 2; 1]` est contenue dans la liste `[2; 1; 2; 1]` mais pas dans la liste `[1; 2]` ni dans la liste `[1; 2; 3]`;
8. `liste_egal liste1 liste2`: teste l'égalité de deux listes (modulo des permutations des éléments de chaque liste);
9. `dernier l`: retourne le dernier élément de la liste `l`;
10. `subst a b l`: substitue l'élément `a` à toute occurrence de l'élément `b` dans la liste `l`;
11. `subst1 a b l`: substitue l'élément `a` à la première occurrence de l'élément `b` dans la liste `l` et seulement à cette occurrence;
12. `renverser l`: retourne une liste contenant les mêmes éléments que `l` mais en ordre inverse.