

## Listes et filtrage

Pour chacune des fonctions qui suivent :

- implanter la fonction en utilisant la conditionnelle, `hd` et `tl` ;
- implanter la fonction en utilisant le filtrage.

et tester les différentes implantations.

1. `length[A](l: list[A]) -> int`  
renvoie la longueur de la liste.
2. `rev[A](l: list[A]) -> list[A]`  
renvoie la liste retournée. La fonction `reversed` est sa propre inverse. Vérifier cette propriété sur des exemples (définir pour cela une fonction correspondant à `rev o rev` et vérifier que c'est l'identité).
3. `concat[A](l: list[list[A]]) -> list[A]`  
renvoie la liste obtenue en concaténant les éléments de la liste donnée en argument. Autrement dit : `concat([l1, l2, ..., ln]) = l1 + l2 + ... + ln`
4. `rev_append[A](l1: list[A], l2: list[A]) -> list[A]`  
retourne la première liste et la concatène à la seconde. Donner encore une autre implantation de cette fonction en utilisant `rev`.
5. `nth[A](l: list[A], i: int) -> A`  
`nth(l, i)` renvoie le *i*-ème élément de la liste *l*. On supposera que  $0 \leq i < \text{len}(l)$ .
6. `last[A](l: list[A]) -> A`  
renvoie le dernier élément de la liste (que l'on supposera non vide). Donner encore deux autres implantations de cette fonction, l'une en utilisant `rev`, l'autre en utilisant `nth`.
7. `take[A](l: list[A], i: int) -> list[A]`  
`take(l, i)` renvoie la liste composée des *i* premiers éléments de *l*. En particulier, on doit avoir `take(l, len(l)) = l`. On supposera que  $0 \leq i \leq \text{len}(l)$ .
8. `drop[A](l: list[A], i: int) -> list[A]`  
`drop(l, i)` renvoie la liste obtenue en enlevant les *i* premiers éléments de *l*. En particulier, on doit avoir `drop(l, len(l)) = []`. On supposera que  $0 \leq i \leq \text{len}(l)$ .
9. `zip[A, B](l1: list[A], l2: list[B]) -> list[tuple[A, B]]`  
combine *l1* et *l2* en une liste de couples : `zip([a1, ..., an], [b1, ..., bn])` renvoie `[(a1, b1), ..., (an, bn)]`. Si les listes ont des longueurs différentes, `zip` ignore les éléments en excès de la liste la plus longue.
10. `unzip[A, B](l: list[tuple[A, B]]) -> tuple[list[A], list[B]]`  
renvoie un couple de listes obtenues en séparant chaque couple de la liste donnée en argument. Cette fonction est l'inverse de la fonction `zip` vue en cours dans le cas où les listes sont de même longueur. Vérifier cette propriété sur des exemples (définir pour cela deux fonctions correspondant à `zip o unzip` et `unzip o zip` respectivement).