

Pour interpréter OCaml, le plus simple est d'utiliser l'interpréteur en ligne <https://try.ocamlpro.com>. Vous pouvez aussi utiliser un IDE avec un interpréteur sur votre ordinateur si vous préférez.

Exercice 1 :**Fonctions récursives élémentaires sur les listes**

1. (a) Écrire une fonction récursive `length : 'a list -> int` qui calcule la longueur d'une liste donnée en argument.
Remarque : ceci correspond à la fonction prédéfinie `List.length`.
(b) Montrer que cette fonction a une complexité linéaire en utilisant une relation de récurrence sur $C(n)$.
2. (a) Écrire une fonction récursive `borne_sup : int list -> int` qui calcule le plus petit entier majorant les termes d'une liste :
 - l'entier `min_int`, de valeur -2^{31} dans l'interpréteur en ligne¹, si la liste est vide :
 - l'entier maximal parmi les éléments de la liste sinon.(b) Prouver par récurrence sur la longueur de la liste la correction de votre fonction (*On initialisera sur 0 et 1*).

Exercice 2 :**Fonction curryfiée**

Écrire les fonctions suivantes de façon curryfiée, en précisant leur type :

1. Une fonction qui calcule le minimum de deux valeurs.
2. Une fonction qui calcule la i^{e} valeur d'une liste l , pour i entier entre 0 et la longueur de la liste (dans le cas contraire, on déclenche une erreur).
3. Une fonction de type `('a -> 'a) -> ('a -> 'a)` qui prend en argument f et renvoie $f \circ f$.

1. Si vous utilisez l'interpréteur normal, ou que vous compilez le programme, c'est -2^{62} . Nous ne rentrerons pas dans les raisons de cette valeur plutôt que la valeur attendue -2^{63} .