

| | |
|----------------|-------------------|
| Nom : | Note : / 10 |
| Prénom : | |

□ **Exercice 1 :** *Représentation des ensembles d'entiers*

En OCaml, on propose de représenter un ensemble d'entiers, par la liste *triée* (dans l'ordre croissant) de ses éléments. Par exemple l'ensemble $\{2; 3; 5; 7\}$ sera représenté par la liste $[2; 3; 5; 7]$. Ainsi une liste d'entiers représente correctement un ensemble lorsque :

- ses éléments sont dans l'ordre croissant,
- et chaque élément figure en un seul exemplaire.

Par exemple, les listes $[2; 3; 3; 5; 7]$ (élément en double) ou $[2; 5; 3; 7]$ (non triée) ne représentent pas correctement un ensemble.

1. Ecrire une fonction `est_ensemble: int list -> bool` qui renvoie `true` lorsque la liste d'entier fournie en argument représente correctement un ensemble.

[illegible]

2. Ecrire une fonction `appartient: int -> int list -> bool` qui prend en argument un entier et une liste (représentant un ensemble) et renvoie `true` lorsque l'entier appartient à l'ensemble représenté par la liste. Par exemple `appartient 3 [2; 3; 5; 7]` renvoie `true`.

[illegible]

3. Ecrire une fonction `union` : `int list -> int list -> int list` qui prend en argument deux listes d'entiers (en supposant que ces deux listes représentent correctement des ensembles) et renvoie la liste d'entiers représentant l'union de ces deux listes. Par exemple `union [2; 5; 7] [5; 6; 7; 10];;` renvoie `[2; 5; 6; 7; 10]`.

[illegible]