

Exercice 1 : Enregistrements

1. Déclarer un type `etudiant` composé par un `nom`, un `prenom` et une liste `l` qui contient les notes de colles.
2. Ecrire une fonction `moyenne` qui calcule la moyenne des notes d'un étudiant.
3. Ecrire une fonction `moyenne_adaptee` qui calcule la moyenne des notes d'un étudiant en ignorant la plus mauvaise. Vous pouvez vous aider d'une fonction `minimum` qui calcule le minimum d'une liste des flottants.
4. Ecrire une fonction récursive `reussite` de profil `list etudiant -> int` qui calcule le nombre des étudiants qui ont réussi la première colle (première note dans la liste `l` d'un étudiant), la moyenne étant 10.

Rappel : quelques fonctions standards sur les listes

<code>List.length</code>	longueur d'une liste
<code>List.hd</code>	premier élément de la liste
<code>List.tl</code>	liste privée de son premier élément
<code>List.map</code>	<code>List.map f [a ; b ; ... ; c]</code> calcule <code>[f a ; f b ; ... ; f c]</code>
<code>List.fold_left</code>	<code>List.fold_left f a [b1 ; b2 ; ... ; bn]</code> calcule <code>f (...(f (f a b1) b2)...) bn</code>

Exercice 2 : Tri sélection

Proposer une implantation en CAML du tri sélection d'une liste. Le procédé est le suivant :

1. Tant que la liste n'est pas vide, on recherche le minimum et on l'insère en tête de la liste résultat.
2. On recommence ce traitement sur la liste initiale dans laquelle le minimum précédemment recherché a été supprimé.

Pour quel type de listes le tri sélection va-t-il effectuer le plus de comparaisons ?

Exercice 3 : Tri fusion

Dans la suite on dira qu'une liste est triée si elle est triée par ordre croissant.

1. Écrire une fonction récursive `coupe` de type `'a list -> ('a list * 'a list)` qui découpe une liste en deux listes de taille égale, à un près.
2. Écrire une fonction récursive `fusion` de type `('a list * 'a list) -> 'a list` qui fusionne deux listes triées en une seule liste triée.
3. En déduire une fonction récursive `tri_fusion` de type `'a list -> 'a list` qui trie une liste donnée en argument.
4. (Question bonus) : calculer le nombre de comparaison effectué par le tri fusion pour trier une liste dans le pire des cas.

Exercice 4 : Construction d'expressions de type donné

Pour chacun des items suivants, donner une expression du même type ; par exemple pour le type `int list` on peut donner l'expression `[1 ;2 ;3]`.

1. `(char*char) list`
2. `(int->float) -> float -> float`
3. `(float -> string) list`
4. `'a -> 'b -> 'a`
5. `'a -> 'a -> bool`