

FEUILLE D'EXERCICES N°6 DE L'OPTION D'INFORMATIQUE.

listes et récursivité.

Le but de ce TD est d'utiliser les listes en CAML ainsi que la récursivité et la reconnaissance de motifs par CAML.

1. Programmer les fonctions suivantes (il s'agit de reprogrammer les fonctions prédéfinies en CAML quand elles existent) :

- qui retourne la tête d'une liste (fonction Caml **hd**),
- qui retourne la queue d'une liste (fonction Caml **tl**),
- qui renverse une liste (fonction Caml **rev**),
- qui calcule le plus grand élément d'une liste de réels,
- qui calcule la longueur d'une liste (fonction Caml **list_length**),
- qui colle deux listes (fonction Caml **@**),
- qui vérifie si un élément appartient à liste (elle retournera le rang de la première occurrence, s'il est dans la liste, -1 sinon)(proche de la fonction Caml **index**),
- fonction permettant de retourner le dernier élément d'une liste,
- fonction qui élimine toutes les occurrences d'un élément x dans une liste,
- fonction qui élimine tous les doublons dans une liste (on ne garde qu'un représentant).
exemple: **doublon [1;2;3;4;1;1;2;5;3] = [1;2;3;4;5]**.
- fonction qui applique une fonction f à tous les éléments d'une liste (fonction Caml **map**)
exemple: **map f [a₁; ...; a_n]** retourne la liste $[f(a_1); \dots; f(a_n)]$,
- fonction qui admet comme paramètres une liste de fonctions $[f_1; \dots; f_n]$ et une liste d'éléments $[a_1, \dots, a_n]$ et qui retourne la liste $[f_1(a_1); \dots; f_n(a_n)]$,
- fonction qui admet comme paramètres une fonction f et une liste $[a_1; \dots; a_n]$ et qui retourne la valeur $f(a_1, f(a_2, \dots f(a_{n-2}, f(a_{n-1}, a_n)) \dots))$ (c'est presque la fonction Caml **list_it**)
appliquer la fonction précédente pour calculer la somme des éléments d'une liste d'entiers, puis au produit des éléments d'une liste de réels,
- fonction qui effectue une permutation circulaire à droite sur une liste,
- fonction qui effectue une permutation circulaire à gauche sur une liste,
- fonction de duplication de liste qui à partir de la liste $[a_1; a_2; \dots; a_n]$ renvoie $[a_1; a_1; a_2; a_2; \dots; a_n; a_n]$
- fonction qui à partir de la liste $[a_1; a_2; \dots; a_n]$ calcule $\prod_{i < j} (a_i - a_j)$,

2. On veut simuler les opérations ensemblistes à l'aide de listes. Un ensemble A sera donc représenté par la liste de ses éléments (ils sont supposés être tous du même type)

- Écrire une fonction **appartient** qui renvoie le booléen *true* si l'élément x appartient à l'ensemble A et *false* sinon.
- Écrire une fonction qui calcule l'intersection de deux ensembles (fonction Caml **intersect**),
- Écrire une fonction qui calcule la réunion de deux ensembles (fonction Caml **union**),
- Écrire une fonction différence ensembliste (fonction Caml **subtract**),
- Écrire une fonction différence symétrique ($A\Delta B = (A \cup B) \setminus (A \cap B)$),
- Écrire une fonction qui étant donné l'ensemble A renvoie l'ensemble $\mathcal{P}(A)$ des parties de A .