Les Listes

1/ Définition de la liste :

Les listes sont des structures permettant de regrouper un nombre quelconque d'éléments de même type. Exemples :

Liste d'entiers	Liste de réels	Liste de chaines de car	Liste de listes d'entiers	Liste vide
[14 ; 5 ; 0 ; 658]	[12.5 ; 25.8]	["ali" ; "omar" ; "kader"]	[[1 ; 6] ; [5] ; [] ; [12 ; 25 ; 8]]	[]
Type : int list	Type : float list	Type : string list	Type : int list list	Type : 'a list

2/ Les Opérations sur les listes :

Deux opérations sont prédéfinies en CAML :

La concaténation de liste, notée **@**	L'ajout d'un élément en tête de liste, notée **::**
# [1 ;5] @ [2;8] ;; - : int list = [1; 5; 2; 8]	# 5 :: [3;8] ;; - : int list = [5; 3; 8]
# [] @ [2;8] ;; - : int list = [2; 8]	# (1 ::(2 ::(3 :: [ ]))) ;; - : int list = [1; 2; 3]

Deux fonctions sont prédéfinies en CAML :

fonction ***hd*** : retourne le 1er élément de la liste	fonction ***tl :*** retourne la liste sans le 1er élément
# hd [12; 25 ; 18] ;; - : int = 12	# tl [12 ;25 ;18] ;; - : int list = [25; 18]

3/ Quelques fonctions utilisant les listes :

# let f x y = x @ y ;; f : 'a list -> 'a list -> 'a list = <fun>	# let g x y = x :: y ;; g : 'a -> 'a list -> 'a list = <fun>
# f [12; 2] [20] ;; - : int list = [12; 2; 20]	# g 22 [12; 2 ; 10] ;; - : int list = [22; 12; 2 ; 10]
# f ["toto";"tata"] ["titi" ; "tutu"] ;; - : string list = ["toto"; "tata"; "titi"; "tutu"]	# g "toto" ["tata" ; "titi" ; "tutu"] ;; - : string list = ["toto"; "tata"; "titi"; "tutu"]

Ecrire en langage CAML les fonctions suivantes :

1. Calcule le nombre d'éléments d'une liste (***nbElm***).
2. Calcule la somme des éléments d'une liste (***somElm***).
3. Calcule la moyenne des éléments d'une liste en utilisant les fonctions ***nbElm*** et ***somElm***.
4. Insère un élément au début d'une liste.
5. Insère un élément à la fin d'une liste.
6. Supprime un élément au début d'une liste non vide.
7. Supprime un élément à la fin d'une liste non vide (utiliser la fonction ***nbElm***).
8. Inverse une liste.
9. Projette l'élément n°I (I >0) d'une une liste non vide.
10. Détermine si un élément donné appartient ou non a une liste.
11. prend un nombre entier et une liste d'entier et compte les occurrences de ce nombre dans cette liste.
12. Calcule la fonction ***Map*** définie par : ***Map f*** [a1 ; a2 ; … ; an] -> [(***f*** a1) ; (***f*** a2) ; … ; (***f*** an)]
    (Distribue la fonction ***f*** sur les éléments de la liste).
    - Appliquer la fonction ***Map*** pour élever au carrée les éléments d'une liste d'entiers.
    - Appliquer la fonction ***Map*** pour inverser les mots dans une liste de mots.
13. Trie une liste d'entiers par ordre croissant.