Les listes

1 Parcours simples

Exercice 1.1 (Somme)

Écrire une fonction qui calcule la somme de tous les éléments d'une liste d'entiers.

Exercice 1.2 (Compte)

Écrire une fonction qui compte le nombre d'occurrences d'une valeur donnée dans une liste quelconque.

Exercice 1.3 (Recherche)

Écrire une fonction qui recherche si un élément est présent dans une liste.

is_empty
head
end
second.

Exercice 1.4 (i$^{\text{ème}}$)

Écrire une fonction qui donne la valeur du $i^{ème}$ élément d'une liste. La fonction devra déclencher une exception `Invalid_argument` si i est négatif ou nul, ou une exception `Failure` si la liste est trop courte.

Exercice 1.5 (Maximum)

Écrire une fonction qui retourne la valeur maximum d'une liste.

2 Le résultat est une liste

Exercice 2.1 (Liste arithmétique)

Écrire la fonction `arith_list` n a_1 r qui construit la liste des n premiers termes de la suite arithmétique de premier terme a_1 et de raison r.

Exemples d'application :

```
# arith_list 12 2 1;;
 - : int list = [2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13]
# arith_list 11 0 3;;
 - : int list = [0; 3; 6; 9; 12; 15; 18; 21; 24; 27; 30]
```

Exercice 2.2 (Sommes cumulées – *C1# - 03/2018*)

Écrire la fonction CAML `sum_cumul` qui, à partir d'une liste d'entiers, construit une liste dans laquelle chaque $i^{ème}$ élément est la somme des i premiers entiers de la liste d'origine.
Exemples d'application : :

```
# sum_cumul [1; 2; 3; 4; 5] ;;
 - : int list = [1; 3; 6; 10; 15]
# sum_cumul [] ;;
 - : int list = []
```