Programmation : TP 13

Objectifs : [OCaml] Listes.

Fonctions de la librairies

Au cours de ce TP, vous devez utiliser certaines fonctions du module `List`. Vous pouvez consulter les fonctions exportées par ce module en ouvrant la documentation en ligne : `http://rafale`, section `Documentation`, `OCaml Reference Manual`, `The standard library` et `Module List`.

Manipulations

1. Écrire une fonction `n_n2` qui prend une liste d'entiers en argument et qui renvoie la liste des couples (élément, élément au carré).

```
#n_n2 [1;2;3;4];;
 - : (int * int) list = [(1, 1); (2, 4); (3, 9); (4, 16)]
```

2. Réécrire la fonction `n_n2` en utilisant la fonction `List.map`.
3. Écrire la fonction `dernier` qui renvoie le dernier élément d'une liste. Traiter le cas vide avec une exception (`failwith`).

```
#dernier [1;2;3];;
 - : int = 3

#dernier [];;
Exception: Failure "dernier: liste vide".
```

4. Écrire une fonction `telque` qui calcule la sous-liste des éléments d'une liste qui vérifient un prédicat unaire p. Réécrire la fonction en utilisant l'itérateur `List.fold_right`.

```
#telque (fun n -> (n mod 2 = 0)) [1;2;3;4;5;6;7];;
 - : int list = [2; 4; 6]
```

5. Écrire la fonction `entiers` qui fabrique la liste des n premiers entiers (dans un ordre quelconque).

```
#entiers 10;;
 - : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]
```

6. Écrire une fonction `produit` qui réalise le produit cartésien de deux listes. Indication : une fonction intermédiaire n'est pas inutile.

```
#produit [1;2;3] ["un"; "deux"];;
 - : (int * string) list =
[(1, "un"); (1, "deux"); (2, "un"); (2, "deux"); (3, "un"); (3, "deux")]
```

7. Écrire la fonction `nieme` qui renvoie le $n$ième élément d'une liste (lever une exception si la liste contient moins de n éléments).

```
#nieme 2 [3;2;1;0];;
 - : int = 2
```