

TP 4 : Algorithmique fonctionnelle

Listes

Hervé de Milleville - Stefan Bornhofen - Peio Loubière

7 février 2013

1 Listes

1.1 Ré-inventons la poudre...

Écrire les fonctionnalités permettant de manipuler les fonctions existantes suivantes :

- =
- `fold_right`
- `find`
- `for_all`

1.2 Innovons !

Écrire les fonctionnalités suivantes :

- `ajoutFin`, telle que : `ajoutFin 'd 1 ['a; 'b; 'c] = ['a; 'b; 'c; 'd]`
- `ajoutieme`, telle que : `ajoutieme 'd 1 ['a; 'b; 'c] = ['a; 'd; 'b; 'c]`
- `repliquer`, telle que : `repliquer ['a; 'b; 'c] 3 = ['a; 'a; 'a; 'b; 'b; 'b; 'c; 'c; 'c];;`
- `ssListe`, telle que : `ssListe ['a; 'b; 'c; 'd; 'e; 'f; 'g; 'h; 'i; 'j] 2 6 = ['c; 'd; 'e; 'f; 'g];;`

1.3 Tiens, ça me rappelle quelque chose...

- Définir un type `etudiant` composé d'un nom, d'un prenom et d'une liste de notes de DS
- Écrire une fonction `moyenne` qui calcule la moyenne des notes d'un étudiant
- Écrire une fonction `moyenneLissee` qui calcule la moyenne des notes d'un étudiant en ignorant la meilleure et la plus mauvaise
- Écrire une fonction `passerait` qui, pour une liste d'étudiants, retourne la liste de ceux qui ont eu la moyenne à leur dernier DS. Le dernier DS étant, évidemment, le dernier de la liste.