

## TD11 : Types algébriques

**Question 1 :** On considère le type

```
type rationnel = int * int (* numérateur et dénominateur *)
```

Implémenter les fonctions en ajoutant les annotations de type nécessaire pour qu'il soit clair qu'on travaille sur des rationnels :

```
est_rationnel : rationnel -> bool
sont_egaux : rationnel -> rationnel -> bool
entier_vers_rationnel : int -> rationnel
multiplication : rationnel -> rationnel -> rationnel
addition : rationnel -> rationnel -> rationnel
```

**Question 2 :** On se propose de faire une anti-sèche sur les conversions d'unités de longueur pour élève de primaire. On crée le type

```
type longueur = Metre of float | Decimetre of float | Centimetre of float
               | Millimetre of float | Decametre of float
               | Hectometre of float | Kilometre of float
```

Écrire une fonction `metre_of_longueur : longueur -> float` qui convertit une longueur donnée en sa valeur en mètres.

**Question 3 :** On se propose maintenant de faire une anti-sèche sur les calculs des limites pour élève de spécialité en Terminale. On crée le type

```
type limite = Zero | Fini of float | PlusInfini | MoinsInfini | Indetermine
```

Proposer le code pour les fonctions `somme`, `produit`, `différence` et `quotient`.

**Question 4 :** Proposer une implémentation en ocaml d'un type `liste` qui permet d'avoir la longueur de la liste en temps constant. Implémenter la fonction `ajout : int -> liste -> liste` qui ajoute un élément dans la liste en temps constant.

**Question 5 :** Au bridge, la phase de jeu est précédée d'une phase d'enchères entre les deux équipes de joueurs. Une *enchère* est l'annonce d'un certain nombre de levées<sup>1</sup> (entre 1 et 7 —l'équipe qui gagne les enchères est supposée faire un nombre de plis au moins égal à 6 plus le nombre de levées annoncé) et d'une couleur (♠, ♥, ♦ ou ♣). Lors de cette phase, un joueur qui propose une enchère ne peut que surenchérir sur la dernière enchère proposée.

Les règles pour surenchérir sont les suivantes, il faut :

- soit annoncer plus de levées (quelles que soient les couleurs concernées),
- soit enchérir avec une couleur supérieure si le nombre de levées est identique.

L'ordre des couleurs est : ♠ > ♥ > ♦ > ♣.

Proposer des types et une fonction qui permettent de tester si une enchère est supérieure à une autre. On conclura avec un `failwith` si l'enchère est irréaliste.

**Question 6 :** On définit le type

```
type point = float * float (* abscisse et ordonnée *)
```

Proposer une fonction ocaml qui permet de calculer l'isobarycentre d'une liste de points, avec un seul parcours de la liste.

---

1. Une *levée* est un pli.