

## Types complexes

### Exercice 1

Le but de cet exercice est de programmer une fonction permettant de calculer l'aire d'un quadrilatère convexe, donné par les coordonnées de ses quatre sommets. La seule formule dont on dispose est la suivante, calculant l'aire d'un triangle donné par les longueurs  $a, b, c$  de ses trois côtés :  $S = \sqrt{p(p-a)(p-b)(p-c)}$  où  $p = (a+b+c)/2$ .

1. Définir le type `point` comme enregistrement avec un champ `abs` pour l'abscisse et un champ `ord` pour l'ordonnée.
2. Décomposer le problème posé à l'aide de deux sous-problèmes que l'on définira.
3. Donner les deux fonctions associées à ces deux sous-problèmes.
4. Écrire finalement la fonction `aire_quadrilatere :point -> point ->point -> point -> float`.

### Exercice 2

Un répertoire téléphonique simple contient pour chaque personne son nom, son prénom et son numéro de téléphone.

1. Définir le type `personne` à l'aide d'un enregistrement.  
Un répertoire sera alors simplement une liste d'enregistrements `personne`.
2. Ecrire une fonction qui recherche dans un répertoire le téléphone d'une personne donnée par son nom et prénom.
3. Ecrire une fonction qui recherche dans un répertoire toutes les personnes ayant un nom donné.
4. Comment optimiser les fonctions précédentes ?
5. Modifier le type `personne` afin de permettre de stocker plusieurs numéros de téléphone par personne.

### Exercice 3

Cet exercice concerne le jeu de tarot qui est constitué de 78 cartes.

On distingue trois types de cartes :

- a) les cartes "ordinaires" qui ont une couleur parmi les 4 suivantes : Pique, Cœur, Carreau, et Trèfle et une valeur pouvant être soit un "honneur" (roi, dame, cavalier ou valet), soit un entier compris entre 1 et 10
  - b) les "atouts" qui ont une valeur entière comprise entre 1 et 21
  - c) une carte spéciale appelée l'"Excuse".
1. Définir le type `carte_ordinaire` par un enregistrement utilisant les types `couleur` et `valeur` à définir également.
  2. Définir le type `carte_tarot`.

3. La valeur des cartes est donnée par :

	Atout
Excuse	4.5
Atout 1	4.5
Atout 21	4.5
Atout 2...20	0.5
Roi	4.5
Dame	3.5
Cavalier	2.5
Valet	1.5
1...10	0.5

4. Ecrire une fonction `évalue_carte` qui prend en argument une carte et renvoie sa valeur.
5. Ecrire une fonction `évalue_jeu` qui prend en argument une liste de cartes et renvoie la somme des valeurs.