

## Feuille 4 : Application : plan complexe

**Exercice 4.1** Dans un plan muni d'un repère orthonormé, on associe au point  $P$  de coordonnées  $(x, y)$  son affixe, le nombre complexe  $z = x + iy$ .

Pour représenter un point du plan, on utilise le type

```
type mycomplex = C of float * float
```

Tous ce qui suit doit être écrit dans un fichier nommé `mycomplex.ml`.

1. Définir le type `mycomplex`.
2. Écrire la fonction constructeur `make_complex` qui fabrique un complexe à partir de sa partie réelle et de sa partie imaginaire
3. Écrire l'accessor `realpart` qui permet de récupérer la partie réelle d'un complexe.
4. Écrire l'accessor `imagpart` qui permet de récupérer la partie imaginaire d'un complexe.
5. En utilisant `make_complex`, définir
  - la variable `c_origin` ayant pour valeur le complexe  $(0., 0.)$  (point origine du plan complexe),
  - la variable `c_i` ayant pour valeur le complexe  $i = (0., 1.)$ ,
  - la variable `p_12` ayant pour valeur le complexe  $(1., 2.)$ .
  - Implémenter les opérations `c_sum c1 c2`, `c_dif c1 c2`, `c_opp`, `c_mul c1 c2`, `c_abs c`, `c_sca lambda c`, `c_exp c` qui permettent respectivement de calculer la valeur absolue d'un complexe, la somme, la différence, l'opposé, le produit de deux complexes, la multiplication d'un complexe par un scalaire (float), l'exponentielle complexe.

Remarque : il faut que le code des opérations reste valable si on change le type `mycomplex` pour un enregistrement<sup>a</sup> par exemple; il faut donc utiliser la fonction `make_complex` et ne pas utiliser le constructeur `C`.

<sup>a</sup>Voir le chapitre "Enregistrements"

**Exercice 4.2** Une transformation  $F$  du plan transforme tout point  $P$  en son image  $P' = F(P)$ . On peut décrire la transformation  $F$  par la fonction  $f$  qui appliquée à  $z$  l'affixe de  $P$  donne  $z'$  l'affixe de  $P'$ .

$$\begin{aligned} f: \mathbb{C} &\rightarrow \mathbb{C} \\ z &\mapsto z' = f(z) \end{aligned}$$

Soit  $O$  le point origine du plan, c'est-à-dire le point d'affixe  $(0., 0.)$ .

1. Soit la translation de vecteur  $\vec{u}$  d'affixe  $z_u$ . Donner l'affixe  $z'$  du transformé d'un point d'affixe  $z$  par cette translation.
2. Donner l'affixe  $z'$  du transformé d'un point d'affixe  $z$  par la rotation de centre  $O$  et d'angle  $\theta$ .
3. Donner l'affixe  $z'$  du transformé d'un point d'affixe  $z$  par la rotation de centre  $C$  d'affixe  $z_c$  et d'angle  $\theta$ ,

- Exercice 4.3** 1. Écrire la fonction `translate c vector` qui donne l'affixe du point obtenu par translation de vecteur `vector:mycomplex` du point d'affixe `c`.
2. Écrire la fonction `rotate0 c angle` qui donne l'affixe du point obtenu par rotation d'angle `angle:float` du point d'affixe `c`.
3. Écrire la fonction `rotate c angle center` qui donne l'affixe du point obtenu par rotation d'angle `angle:float` du point d'affixe `c` autour du point d'affixe `center`.

**Exercice 4.4** Tester vos fonctions avec le fichier `test_mycomplex.ml`.