

TP : Kotlin, prise en main**Exercice 1:**

1.1 Écrire une fonction factorielle de façon itérative puis récursive.
Vous devez par exemple trouver 479001600 pour 12 !

1.2 Regarder la méthode de réduction `fold()` et réécrire la factorielle avec `fold`.

Exercice 2 :

2.1 Regarder la classe `FloatArray` et réécrire l'exercice du cours « Écrire un fonction qui calcule la somme d'un tableau passé en paramètre. Utiliser cette fonction dans un `main` pour calculer la somme de 6 valeurs réelles entrées au clavier. »

2.2 Regarder la fonction `reduce()` et l'utiliser pour écrire la fonction `somme()`.

Exercice 3 :

On représente la forme algébrique d'un nombre complexe sous forme de `String`, par exemple « 2+i3 ».

Écrire les fonctions suivantes :

- `im` : qui renvoie la partie imaginaire d'un nombre complexe
- `re` : qui renvoie la partie réelle d'un nombre complexe
- `module` : qui renvoie le module d'un nombre complexe soit pour $x+iy$: $\sqrt{x^2 + y^2}$
- `argument` : qui renvoie l'argument principal d'un nombre complexe
si la partie réelle vaut 0, l'argument vaut $\pm\pi/2$ en fonction du signe de la partie imaginaire
sinon l'argument vaut $\arctangente(y/x)$
- `addC` : qui ajoute 2 nombres complexes $(x1+x2)+i(y1+y2)$
- `mulC` : qui multiplie 2 nombres complexes $(x1*x2-y1*y2)+i(x1*y2+x2*y1)$

par exemple :

1+i2 a pour module 2.23606797749979
et pour argument 1.1071487177940904

`addC("1+i2", "2+i1")` donne **3.0+i3.0**
`mulC("1+i2", "2+i1")` donne **0.0+i5.0**