

Exercice 1 (patron)

Créer une fonction permettant de calculer le carré d'une valeur de n'importe quel type. Tester sur différents types, `int`, `double`.

Tester sur le type `Fraction` du TP n°5.

Exercice 2 (patron)

Créer une classe `Paire` permettant de stocker deux valeurs d'un type donné (`Paire<int>` ou `Paire<Lunette>`, etc). Son usage pourrait ressembler à :

```
Paire<int> p(4,5);  
cout << p.get1() << endl;  
cout << p.get2() << endl;  
p.set1(11); // stocke 11 comme première valeur  
p.set2(22); // stocke 22 comme second élément de la paire
```

Créer une fonction permettant de calculer, à partir d'un `vector` d'éléments la paire constituée du plus grand et du plus petit des éléments. Tester sur avec un vecteur d'entiers, un vecteur de fractions, etc.

Ajouter de quoi afficher une paire, par exemple en séparant les valeurs par `:` et en entourant le tout des caractères `{}` comme `{ 11 , 22 }`

Exercice 3 (patron)

Définir un type `Vecteur` permettant de spécifier la dimension de l'espace auquel ce vecteur appartient ainsi que le type de représentation des coordonnées du vecteur:

```
Vecteur<double,3> v("v");  
v[0] = 12.5;  
v[1] = 23.5;  
v[2] = 3;  
cout << v << endl; // affiche: v=(12.5,23.5,3)  
Vecteur<double,2> v2("w");  
v2[0] = 12.5;  
v2[1] = 23.5;  
cout << v2 << endl; // affiche: w=(12.5,23.5)
```

Essayez avec un `Vecteur<Vecteur<int,3>,2>...`