

Exercice 69

Énoncé

On souhaite réaliser une classe `vecteur3d` permettant de manipuler des vecteurs à trois composantes. On prévoit que sa déclaration se présente ainsi :

```
class vecteur3d
{   float x, y, z ;           // pour les 3 composantes (cartésiennes)
    .....
} ;
```

On souhaite pouvoir déclarer un vecteur, soit en fournissant explicitement ses trois composantes, soit en en fournissant aucune, auquel cas le vecteur créé possédera trois composantes nulles. Écrire le ou les constructeurs correspondants :

- a. en utilisant des fonctions membre surdéfinies ;
- b. en utilisant une seule fonction membre ;
- c. en utilisant une seule fonction en ligne.

Exercice 70

Énoncé

Soit une classe `vecteur3d` définie comme suit :

```
class vecteur3d
{   float x, y, z ;
    public :
        vecteur3d (float c1=0.0, float c2=0.0, float c3=0.0)
        { x = c1 ; y = c2 ; z = c3 ;
        }
        .....
} ;
```

Introduire une fonction membre nommée `coincide` permettant de savoir si deux vecteurs ont les mêmes composantes :

- a. en utilisant une transmission par valeur ;
- b. en utilisant une transmission par adresse ;
- c. en utilisant une transmission par référence.

Si `v1` et `v2` désignent 2 vecteurs de type `vecteur3d`, comment s'écrit le test de coïncidence de ces 2 vecteurs, dans chacun des 3 cas considérés ?

Exercice 71

Énoncé

Soit une classe `vecteur3d` définie comme suit :

```
class vecteur3d
{
    float x, y, z ;
    public :
        vecteur3d (float c1=0.0, float c2=0.0, float c3=0.0)
        {
            x = c1 ; y = c2 ; z = c3 ;
        }
        .....
} ;
```

Introduire, dans cette classe, une fonction membre nommée `normmax` permettant d'obtenir, parmi deux vecteurs, celui qui a la plus grande norme. On prévoira trois situations :

- le résultat est renvoyé par valeurp ;
- le résultat est renvoyé par référence, l'argument (explicite) étant également transmis par référencep ;
- le résultat est renvoyé par adresse, l'argument (explicite) étant également transmis par adresse.

Exercice 72

Énoncé

Réaliser une classe `vecteur3d` permettant de manipuler des vecteurs à 3 composantes (de type `float`). On y prévoira :

- un constructeur, avec des valeurs par défaut (0),
- une fonction d'affichage des 3 composantes du vecteur, sous la forme :

```
< composante1, composante2, composante3 >
```

- une fonction permettant d'obtenir la somme de 2 vecteurs ;
- une fonction permettant d'obtenir le produit scalaire de 2 vecteurs.

On choisira les modes de transmission les mieux appropriés. On écrira un petit programme utilisant la classe ainsi réalisée.

Exercice 73

Énoncé

Comment pourrait-on adapter la classe `point` créée dans l'exercice 66, pour qu'elle dispose d'une fonction membre `nombre` fournissant le nombre de points existant à un instant donné ?