**ECOLE DES HAUTES ETUDES D'INGENIERIE D'OUJDA**

**ANNEE DE FORMATION : 2019– 2020**

**GENIE INFORMATIQUE: 3EME ANNEE**

Unité de formation : Langage C++ TP10 Formateur : A.Serghini

**Exercice 1:**

On souhaite réaliser une classe vecteur 3d permettant de manipuler des vecteurs de trois composantes. On prévoit que sa déclaration se présente ainsi:

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

On souhaite pouvoir déclarer un vecteur, soit en fournissant explicitement ses trois composantes, soit en fournissant aucune, auquel cas le vecteur créé possédera trois composantes nulles.

Ecrire le ou les constructeur(s) correspondant(s):

- En utilisant des constructeurs surchargés
- En utilisant un seul constructeur.
- En utilisant un seul constructeur en ligne.

**Exercice 2:**

Soit une classe vecteur 3d 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;
                                        }
                .........
         };
```

I- Introduire une fonction membre nommé coïncide permettant de savoir si deux vecteurs ont mêmes composantes:

- En utilisant une transmission par valeur,
- En utilisant une transmission par adresse,
- En utilisant une transmission par référence.

Si v1 et v2 désignent deux vecteurs de type vecteurs3d comment s'écrit le test de coïncidence de ces deux vecteurs, dans chacun des trois cas considérés?

2- Introduire une fonction membre nommé norm_max permettant d'obtenir, parmi deux vecteurs, celui qui a la grande norme. On prévoira trois situations:

- En utilisant une transmission par valeur,
- En utilisant une transmission par référence.
- En utilisant une transmission par adresse.

3- Introduire une fonction membre nommé Vect_Oppose permettant d'obtenir l'opposé d'un vecteur 3d. On prévoira trois situations:

- En utilisant un retour par valeur,
- En utilisant un retour par référence.
- En utilisant un retour par adresse.

**Exercice 3:**

Ecrire un programme en C++ permettant de gérer les chaînes de caractères en développant les méthodes suivantes:

- Constructeur par défaut ;
- Constructeur qui permet d'initialiser une chaine de caractères ;
- Destructeur qui permet de détruire la chaîne créée ;
- Une méthode pour afficher la chaîne ;
- Une méthode pour calculer la longueur de la chaîne ;
- Une méthode pour ajouter un caractère dans une position donnée ;
- Une méthode pour tester l'appartenance d'un caractère à une chaîne;
- Une méthode pour méthode inverser une chaîne;
- Une méthode pour supprimer un caractère d'une chaîne ;

On donnera un exemple d'utilisation avec une méthode main() .