

TP n°1 : Rappel sur langage C++

Objectif :

Ce TP représente un rappel sur les structures de contrôle, les fonctions, les tableaux et les fichiers.

Enoncé :

Partie I- utilisation des fonctions :

1. Ecrire une fonction « Saisie » qui saisit un tableau de réel « T », la saisie s'arrête lorsque l'utilisateur introduit 0. La fonction retourne la taille du tableau.
2. Ecrire une fonction « Recherche » qui prend en paramètre un tableau de réel « T » et un réel « x » et affiche le nombre d'occurrence de « x » dans « T ».
3. Ecrire une fonction « Max_Min » qui recherche le maximum et le minimum d'un tableau « T » passé en paramètre.
4. Dans le programme principal :
 - a. Saisir un tableau en utilisant la fonction saisie.
 - b. Demander à l'utilisateur un réel « x » et afficher le nombre de fois qu'il existe dans « T » en utilisant la fonction « Recherche ».
 - c. En utilisant la fonction « Max_Min », afficher le maximum et le minimum du tableau « T »

Partie II- La programmation C++ sous forme de projets multi-fichiers :

Travail demandé : Refaire le travail demandé dans la partie 1 mais sous forme de projet multi-fichiers. En fait, pour éviter d'écrire dans le programme principal les différentes fonctions qui vont être utilisées, on utilise un fichier d'entête (fichier .h) qui contiendra la déclaration de tous les entêtes de fonctions. Ce fichier sera alors inclus dans les différents fichiers .cpp

TP N° 2 (classes et objets)

Objectif :

Manipuler des formes géométriques basées sur la classe Point.

Enoncé :

Soit la classe Point qui contient deux attributs et les méthodes suivantes ainsi que le programme principal exploitant cette classe:

```
class Point
{
private:
double X;
double Y;
public:
Point(double x = 0, double y = 0);
void Afficher();
double GetX();
double GetY();
void SetX(double x);
void SetY(double y);
};

//programme principale
#include <iostream>
#include "Point.h"
int main( void )
{
cout << " ##### TP1 ##### " << endl;
Point pt1(1,1);
Point pt2;
Point * _pt;
pt2 = Point(3,1);
pt = new Point();
cout << "Position de pt1 : ";
pt1.Afficher();
pt1.SetX( 2 ); pt1.SetY( 3 );
pt1.Afficher();
cout << endl;
cout << "Position de pt2 : ";
pt2.Afficher();
cout << endl;
pt = pt1.Add(pt2);
cout << "Position de pt3 : ";
pt->Afficher();
cout << endl;
delete pt;
return 0;
}
```

Travail demandé :

- 1- Ecrire les fichiers nécessaires pour l'exécution du programme principale cité ci-dessus.
- 2- Modifier la classe Point pour ajouter la fonction membre « Distance » qui calcule la distance entre deux points : c'est-à-dire entre le point courant et un point passé en paramètre à la fonction. Tester cette méthode dans le programme principale.

3- Créer une classe Triangle, permettant de définir un triangle dans le plan à partir de 3 objets Point. Donner à cette classe les fonctionnalités suivantes :

- Les fonctions d'accès en lecture et écriture aux champs (les 3 points).
- Affichage : pour afficher les coordonnées des 3 points.
- Périmètre : qui retourne le périmètre du triangle.
- Translation : qui translate le triangle par un vecteur (utiliser un objet Point pour le vecteur).
- Aire : qui retourne l'aire du triangle (formule de Héron ou $\frac{1}{2}$ périmètre).

$$Aire = \sqrt{p.(p-a).(p-b).(p-c)} \quad \text{avec } p = (a+b+c)/2$$

Avec a, b et c les longueurs des cotés