

TP 1 C++

Exercice 1

Définir un pointeur sur un entier, nommé **p**, un pointeur sur un entier constant, nommé **q**, un pointeur constant sur un entier, nommé **r**. Faites des allocations avec l'opérateur **new**. Que constatez-vous ?

Définir un tableau de dix pointeurs sur des entiers. Faites une allocation de l'ensemble et une désallocation. Vérifiez l'état de la mémoire. Définir une référence sur un entier, nommée **s**. Essayez toutes les affectations entre **p**, **q**, **r** et **s** et expliquez les résultats obtenus.

Exercice 2

On souhaite donner le même nom à trois fonctions. La première additionne deux entiers (type **int**), la deuxième deux réels (type **float**) et la troisième deux tableaux de dix entiers. Ecrivez ces fonctions. Montrer que les appels sont correctement réalisés. Question subsidiaire : Que se passe-t-il lorsqu'un appel est fait avec comme arguments deux **short**. Pourquoi ?

Exercice 3

On déclare trois nombres réels **a**, **b**, **c** (**float**) dans le **main()**, et on leur attribue des valeurs arbitraires, par exemple **a=10.5**, **b=-5.3** et **c=-0.2**. On désire écrire une fonction **ordonnetrois** (à définir) qui puisse manipuler ces trois variables passées en paramètre et les ordonner par ordre croissant en permutant leurs valeurs. Afin de simplifier le problème, on décide d'écrire d'abord une fonction **ech2paradresse()** qui effectue le tri sur seulement deux paramètres à la fois, et dont on se sert pour ensuite écrire **ordonnetrois()**. Dans cet exercice, vous utiliserez lorsque c'est nécessaire un passage de paramètres par adresse. On reprend la même question mais on décide d'utiliser des passages par référence.

Exercice 4

Lire l'énoncé jusqu'à la fin avant de commencer.

Créer une classe **Tableau**. Le programme devra demander à l'utilisateur combien de valeurs sont à traiter, il mémorise ensuite chaque valeur (ce seront des réels simple précision : **float**). La classe comprend mis à part les constructeurs une méthode **moyenne()** pour calculer la moyenne de ces valeurs et l'afficher. Le programme propose ensuite de supprimer une des valeurs du tableau. L'utilisateur tape une valeur numérique qui est transmise à une méthode **enleveelement()** qui supprime du tableau la première occurrence de cette valeur rencontrée dans le tableau.

On considère qu'ici "supprimer" consiste à décaler les éléments suivants du tableau d'un cran. Ainsi, la valeur disparaît du tableau. Si la fonction parvient à trouver la valeur indiquée par l'utilisateur (et donc à l'enlever), elle renvoie la valeur booléenne **true** et sinon la valeur **false**. En fonction du résultat de la fonction, soit on recalcule la nouvelle moyenne une fois la valeur supprimée, soit on affiche un message signalant que la valeur ne faisait pas partie des échantillons et que la moyenne n'a donc pas changé.

Exercice 5

Réalisez une classe `Equation2` permettant, comme son nom l'indique, de représenter et de traiter informatiquement des équations du second degré. La classe devra permettre de saisir et d'afficher les paramètres d'une équation du second degré ainsi que de calculer les solutions de l'équation.

Exercice 6

Ecrire une classe `Polynome`, qui contient les paramètres d'un polynôme d'un degré n . Il serait intéressant de pouvoir réaliser deux autres opérations : intégrer et dériver. Ces opérations créeront à partir d'un polynôme existant, un nouveau polynôme résultat. A vous de proposer une déclaration de classe.