Cahier de TP n°2

TP 4 : Manipulation de tableaux / boucles

Saisie / traitement de valeurs

- Ecrire un programme qui fait la saisie de valeurs de type réel, qui les range dans un tableau et qui effectue les calculs suivants :
  a) s'il y a plus de 3 valeurs : calcul de la somme des valeurs absolues
  b) s'il y a exactement 2 valeurs : calcul de leur moyenne harmonique $\overline{h}$:

$$\frac{2}{\overline{h}} = \frac{1}{x_1} + \frac{1}{x_2}$$

  où x_1 et x_2 sont les deux valeurs du tableau.
  c) s'il y a exactement 4 valeurs : calcul de leur moyenne géométrique $\overline{g}$:

$$\overline{g} = \sqrt[4]{x_1.x_2.x_3.x_4}$$

  où x1,…x4 sont les quatre valeurs du tableau.

  Pour le calcul de la racine quatrième : calculer la racine quatrième revient à élever le nombre à la puissance ¼, soit 0,25. On utilisera une instruction de calcul : *pow(x,y)* qui calcule le nombre *x* (réel) élevé à la puissance *y* (nombre réel), et qui donne un résultat réel. Pour cela, il est nécessaire d'ajouter la ligne de directive : `#include <math.h>` au début du programme.

- Ecrire un programme qui recherche le minimum et le maximum parmi un tableau de nombres entiers initialisé (l'initialisation peut être faite par une saisie ou manuellement à l'intérieur du programme).

Mise à l'échelle (scaling) de valeurs réelles contenues dans un tableau

Considérons un tableau X contenant un certain nombre de valeurs x_i de type reel (stockant des nombres à virgule). On souhaite les transformer pour aboutir à un deuxième tableau Y contenant des valeurs y_i. Le principe de mise à l'échelle est d'appliquer à toutes les valeurs x_i stockées dans le tableau X la même transformation linéaire $y_i = a.xi+b$, afin que toutes les valeurs y_i soient comprises entre deux bornes y_{min} et y_{max} définies à l'avance. Cela permet, par exemple, si le tableau X contient les valeurs d'un signal électronique échantillonné, de s'affranchir de l'amplitude de ce signal. Pour l'instant, on considère que

y_{min}=0.0 et que y_{max}=1.0. Il est aisé d'en déduire que toutes les valeurs stockées dans le tableau *Y* seront comprises entre 0 et 1.

a) comment calculer les paramètres *a* et *b* de la transformation linéaire à partir des valeurs x_i stockées dans le tableau X ? Pour répondre à cette question, essayez de résoudre un simple système de deux équations à deux inconnues : les 2 équations formant ce système peuvent être facilement trouvées en répondant aux questions suivantes : quelle valeur x_i du tableau *X* sera transformée en 0 ? Quelle valeur x_j du tableau *X* sera transformée en 1 ?
b) Connaissant la valeur des paramètres *a* et *b*, appliquez à toutes les valeurs du tableau *X* la transformation $a.x_i+b$ et rangez les valeurs obtenues dans le tableau *Y*. Affichez les valeurs contenues dans les tableaux *X* et *Y* pour vérifier que la mise à l'échelle a bien fonctionné.
c) Si l'on veut, à partir des valeurs stockées dans *Y*, faire la transformation inverse (revenir aux valeurs du tableau *X*), quelles informations faut-il garder ?
d) Améliorer le programme pour que la mise à l'échelle se fasse entre deux bornes y_{min} et y_{max} saisies par l'utilisateur, au lieu de 0 et 1.

Loi de Titius-Bode et position des planètes

Vers les années 1770, on ne connaissait dans le système solaire que 6 planètes (par ordre de proximité au soleil) : Mercure, Venus, la Terre, Mars, Jupiter et Saturne. Leurs distances respectives au soleil, en fonction de la distance Terre-Soleil (nommée unité Astronomique ou UA, et qui vaut à peu près 150 millions de km) étaient également connues :

- Mercure : 0,387
- Venus : 0,723
- Terre : 1
- Mars : 1,523
- Jupiter : 5,202
- Saturne : 9,554

Deux astronomes de l'époque, Johann Daniel Tietz dit Titius et Johann Elert Bode ont cru déceler une loi mathématique simple permettant d'associer à chaque entier, en partant de 0 (sauf pour le cas de Mercure), le nombre d'UA séparant une planète du soleil. La planète Mercure est associée au nombre $-\infty$, Vénus au nombre 0, etc… La loi qu'ils ont rédigée est la suivante :

Une planète associée au numéro *N* est à une distance moyenne de $(3x2^{N}+4)/10$ UA du soleil (en considérant que $2^{-\infty} = 0$).

En utilisant cette formule, calculez et rangez dans un tableau les distances calculées par cette loi pour *N* variant entre 0 et 5. N'oubliez pas de traiter par un calcul manuel le cas de Mercure. Affichez les différents résultats obtenus sous une forme lisible. Que constatez-vous ?

En 1781, une nouvelle planète est découverte : il s'agit d'Uranus, qui porte le numéro 6. Ajoutez la valeur correspondante dans le tableau.

La distance réelle de cette planète au soleil est de 19,2 UA. La loi de Titius-Bode est elle encore crédible à ce stade ?

Au début du XIXème siècle, de nouveaux corps célestes, situés entre Mars et Jupiter, ont été découverts. Il s'agit de la ceinture d'astéroïdes dont la distance moyenne au soleil est de 2,766 UA. Quelle conclusion en ont tiré les astronomes de l'époque sur l'origine de cette ceinture d'astéroïdes ?

Des perturbations significatives dans l'orbite d'Uranus ont été détectées au début du XIXème siècle également, ce qui a mené à la découverte de Neptune par Urbain le Verrier, et Pluton a été découverte en 1930 par Clyde Tombaugh, lors d'une campagne d'observation systématique du ciel. Leurs distances respectives au soleil sont de : 30,1 UA et 39,44 UA.

A votre avis, la loi de Titius-Bode a-t-elle encore cours aujourd'hui ?

TP5 : Exercices avancés sur les tableaux

Rendement d'un placement

Les rendements obtenus par le CAC40 sur cinq années, de 1998 à 2002, ont été respectivement les suivants : +31%, +51%, -1%, -22% et -34%. Le rendement moyen d'un placement en épargne est de 1,3 %, et n'est pas modifié d'une année sur l'autre.

On utilisera, dans cet exercice, des tableaux :

- Un tableau pour stocker les progressions du CAC40 (dont les valeurs vous sont fournies)
- Un tableau pour stocker la progression d'un capital investi sur un produit indicé sur le CAC40
- Un tableau pour stocker la progression d'un capital investi sur un produit d'épargne

Partie I : placement en épargne

On souhaite calculer le rendement global du produit d'épargne. pour cela, on procède de deux manières :

- On calcule le taux global que l'on applique une seule fois au capital
- On calcule la progression du capital année après année (par accumulation)

Ecrivez un programme qui fait ces deux calculs et faites-le fonctionner avec plusieurs valeurs de capital initial. Relevez les différentes valeurs et remplissez le tableau suivant :

Capital initial	Capital final par calcul global	Capital final par accumulation

Partie II : placement en bourse

On vous demande d'évaluer le rendement moyen d'une somme investie dans un portefeuille d'actions indicé sur le CAC 40. Ecrire un programme qui vous permet de saisir puis d'afficher ces taux de progression.

On souhaite maintenant comparer les deux types de placement : à la suite du programme déjà écrit, calculez et affichez le taux moyen du rendement du placement CAC40

Pour vérifier si ce placement est intéressant, on souhaite visualiser la progression du capital investi à l'aide du taux global calculé précédemment Pour cela, on utilise les deux méthodes vues pour le placement en épargne, mais en utilisant cette fois-ci le taux global associé au placement en bourse : remplissez le tableau suivant.

Capital initial	Capital final par calcul global	Capital final par accumulation

Partie III : vérification de calculs

Quelle que soit la méthode de calcul appliquée (par taux global ou par accumulation), on devrait trouver le même capital final, c'est à dire que les deux dernières colonnes du tableau I devraient être identiques, et les deux dernières colonnes du tableau II devraient être identiques. Si c'est le cas, bravo, vous avez trouvé la bonne formule pour le calcul du taux global, sinon, il faut vos repenchez sur cette formule (mais ne vous inquiétez pas, la confusion est classique)

Attention : le paragraphe précédent ne signifie pas que les deux tableaux doivent comporter les mêmes valeurs, car les taux d'intérêt sont différents !

Ecrivez un programme qui utilise la bonne formule pour le taux global, et faites le fonctionner : vous devez avoir les mêmes résultats avec les deux méthodes.

Plutôt que d'appliquer le taux moyen pour le placement en bourse, on va calculer année après année l'évolution du capital avec le pourcentage réel d'évolution du CAC40 (c'est à dire, +31 % la première année, +51% la deuxième, etc…)

Calculez cette évolution et rangez les valeurs obtenues dans le tableau prévu à cet effet. Que constatez-vous ?

En fait, la formule correcte pour le calcul du taux moyen d'augmentation d'un capital sur plusieurs années est la moyenne géométrique, et non la moyenne arithmétique.

Calculez et affichez la moyenne géométrique des taux d'intérêt

- Pour le placement en bourse
- Pour le placement de type épargne

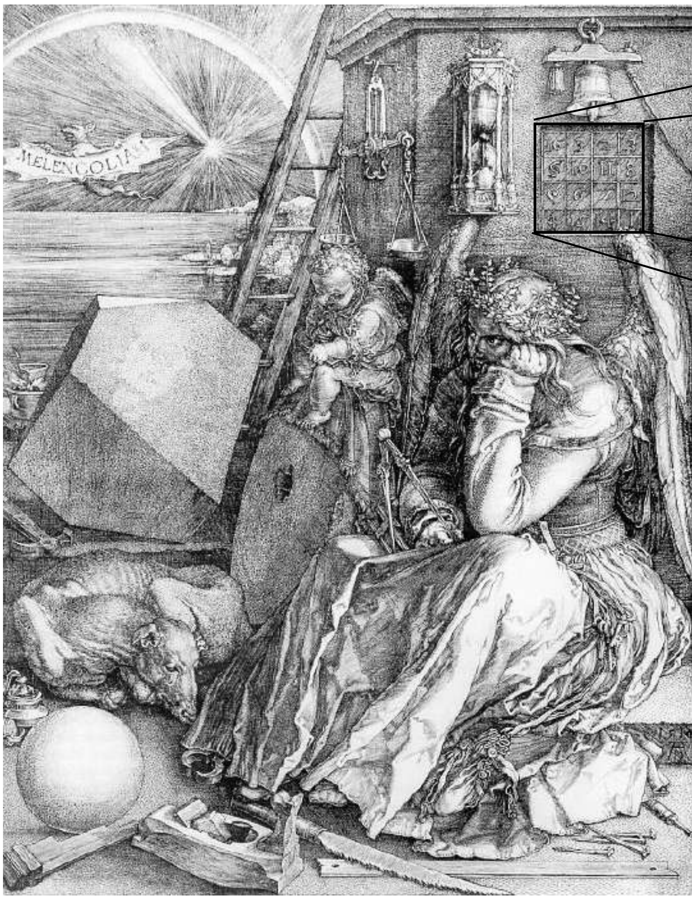
Quel placement vaut-il mieux choisir ?

Tableaux à plusieurs dimensions

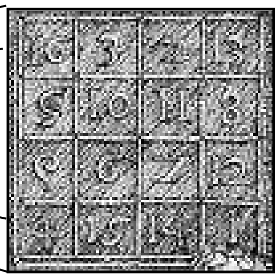
Création d'un carré magique

Un carré magique d'ordre *N* est un tableau à deux dimensions, carré. Cela signifie que la taille utile des deux dimensions est la même. Dans ce tableau seront stockées tous les entiers compris ente 1 et N^2. Cependant, ces valeurs sont rangées de telle sorte que la somme des nombres rangés dans chaque ligne, chaque colonne, et chaque diagonale est la même.

Cet objet mathématique a longtemps fasciné les mathématiciens et les artistes à l'époque de la renaissance, à tel point que le célèbre artiste Albrecht Dürer (1471 – 1528) en a fait apparaître un sur l'une de se gravures nommé Melancholia.



Melancholia (Albrecht Dürer – 1514)



16	3	2	13
5	10	11	8
9	6	7	12
4	**15**	**14**	1

Dürer a même réussi à faire apparaître la date dans le carré magique !

Il existe un algorithme simple de création de carré magique pour les ordres *N* impairs.

Phase initiale :

Dans un premier temps, il est nécessaire d'initialiser toutes les cases du carré magique avec la valeur 0, cette dernière permettant de repérer qu'une case est libre ou déjà occupée par un nombre qui y serait rangé.

Il faut placer le chiffre 1 dans la case située juste au dessus de la case milieu du carré (cette case existe car le carré est d'ordre *N* impair). Si `taille` est la variable contenant la taille utile des deux dimensions du tableau, quels sont les indices de cette case où l'on commence le remplissage du carré magique ?

Phase de remplissage:

On continue en plaçant les nombres 2, 3, …, jusqu'à N^2, selon la diagonale de direction : vers le haut et vers la droite. Soient deux variable `lig` et `col` contenant les indices de la ligne et de la colonne de la case en cours de traitement. Quelles opérations appliquer à `lig` et `col` pour traduire : "vers le haut" et "vers la droite" ? Aidez-vous d'un schéma au besoin.

Bien entendu, par cette progression, on dépasse assez rapidement les bords du carré magique. Quelles sont les conditions à écrire pour tester que les indices `lig` et `col` dépassent du tableau ?

- Si l'indice `lig` dépasse par le haut, alors on repart de la dernière ligne du tableau, en conservant la même valeur pour `col`.
- Si l'indice `col` dépasse par la droite, on repart de la première colonne du tableau, en conservant la même valeur pour `lig`.
- Si l'indice `col` dépasse par la gauche, on repart de la dernière colonne du tableau, en conservant la même valeur pour `lig`.

Vérification de la case atteinte :

Il peut arriver que l'on atteigne une case contenant un nombre rangé précédemment. Comment tester si une case est déjà occupée par un nombre ?

Dans le cas où on atteint une case occupée, on se déplace dans la diagonale vers le haut et vers la gauche (et non plus vers la droite) tant que la case atteinte n'est pas libre. Il faut également contrôler, lors de ce déplacement, que l'on ne dépasse pas les limites du tableau en appliquant les mêmes règles que précédemment. Dès que l'on a réussi à placer le nombre, la progression reprend dans la diagonale vers le haut et vers la droite.

Ecrivez un programme qui remplit un carré magique d'ordre N impair et qui affiche ce carré une fois rempli. Le programme devra vérifier que N est impair (si N est pair, on ne remplit pas le carré), et que $N < 20$ (au delà de N=20, le carré devient difficile à visualiser sur l'écran). Vous vérifierez sur un ou deux exemples que vous obtenez bien un carré magique.

(énoncé tiré de : http://www.infres.enst.fr/~premiere/TP_algo/carre-magique.html)

Le tas de cubes

On cherche à visualiser la dispersion d'un certain nombre de cubes tombant dans des colonnes en utilisant un tableau à deux dimensions. Le principe est simple : on laisse 'tomber' un cube dans un tableau de n lignes et p colonnes, sachant que sa position de départ (x,y) est toujours la même : en haut et au milieu du tableau, soit $x = p/2$ et y=0. On procède ensuite par itération pour matérialiser la chute : à chaque itération, le cube descend d'une ligne, et se dirige, au hasard, d'une colonne vers la gauche ou vers la droite (ce tirage doit être fait à chaque itération). La chute du cube s'arrête lorsqu'il arrive à une position en bas du tableau, ou lorsqu'il y a un autre cube en dessous de la position où il se trouve (c'est-à-dire sur la ligne suivante, à la même colonne).

Si un cube doit se déplacer vers la gauche lors de sa descente et que la case 'en dessous à gauche' est déjà occupée, alors il descend 'tout droit' (même colonne). Idem pour la droite.

Avant de passer à la programmation, faites des schémas pour matérialiser les déplacements d'un cube.

Ecrivez un programme qui montre la chute de 150 cubes dans un tableau de 15 lignes et 30 colonnes. (on devra visualiser étape par étape la chute des cubes).

Pour avoir une idée du résultat attendu, regardez le programme : http://perso.efrei.fr/~flasque/promo2010/Algo_et_C/TP/cubes/cubes.exe.