

Objectifs de la séance: les mêmes que le TD 9, plus:
manipulation de listes ou d'ensembles
calculs « d'effectifs de valeurs d'un caractère », et pourcentages
tri shaker

Exercice 1

- a) Etudier l'exemple `fusion_liste.c` donné dans le dossier TD10 contenant aussi le présent énoncé.
(rappel: l'adresse du site est `http://infolicencemass.univ-paris1.fr`). Vous pouvez le compiler et l'exécuter.
Expliquer chacune des déclarations de l'entête de la fonction `fusion_liste`.
Faire tourner la fonction `fusion_listes` à la main avec les listes $L1=\{1,3,5,7\}$ et $L2=\{2,3,4,5\}$.
Expliquer pourquoi il est indispensable que $L1$ et $L2$ soient tous les deux en ordre croissant.
- b) Ecrire une fonction d'entête `void union_ensembles(int V1[], int V2[], int V3[], int n1, int n2, int *n3)`.
Contrairement au a), les vecteurs $L1$ et $L2$ représentent des ensembles où chaque valeur ne doit figurer qu'une seule fois. En conséquence, le vecteur $V3$ généré devra représenter l'ensemble UNION des 2 ensembles précédents, et chacune de ses valeurs ne devra figurer qu'une seule fois. Autrement dit, il ne doit pas y avoir de doublons dans $V3$. De même, la valeur de l'entier pointé par $n3$ doit être modifier pour donner la cardinalité de l'UNION.
Modifier le programme précédent pour tester cette nouvelle fonction.
- c) Ecrire une fonction d'entête `void inter_ensembles(int V1[], int V2[], int V3[], int n1, int n2, int *n3)` qui effectue l'INTERSECTION des 2 ensembles représentés par $L1$ et $L2$.

Exercice 2 : effectif et fréquence des valeurs d'un caractère.

On est souvent amené à faire des traitements sur des données excédentaires: c'est un des aspects de cet exercice. le fichier `textes/personnes.txt` contient des données sur des personnes. Les données relatives à une personne sont regroupées sur une seule ligne: les 20 premiers caractères contiennent le nom et le prénom, la suite représente la date de naissance codée sous la forme `jj/mm/aa`.

Le traitement qui nous intéresse consiste à calculer, sur l'ensemble de notre population, l'effectif des naissances pour chaque mois de l'année. Deux problèmes se posent alors:

- ignorer les données « inutiles » pour ce traitement: c'est le rôle du format `%*20c` de « sauter » 20 caractères. On pourrait faire de même pour les autres données inutiles...
 - Compter, pour chaque mois de l'année, le nombre de naissances toutes années confondues. A noter que ce type de calcul est parfois désigné sous le terme de « tri à plat », bien que cela soit très éloigné du tri.
- a) Compiler et exécuter le programme `effectif_frequence.c` avec les données du fichier `personnes.txt`. (Les 2 fichiers sont dans le dossier TD10)
- b) Etudier le programme `effectif_frequence.c` donné en exemple, le faire tourner à la main.

Exercice 3 : tri shaker.

- a) Etudier la variante du tri à bulle `tri_bulle1.c` donnée dans le dossier TD10. En particulier, l'usage des variables `D` et `k`.
- b) La fonction `genere_alea` est utilisée pour générer les données à trier de façon pseudo aléatoire. La fonction `srand` sert à initialiser une suite de nombres entiers pseudo aléatoire restituée ensuite, au coup par coup, par la fonction `rand`; ces 2 fonctions appartiennent à `stdlib.h`. La fonction `time` donne, d'une façon qui ne nous intéresse pas ici l'heure du système: elle est utilisée en paramètre de `srand`, pour que la suite pseudo-aléatoire générée soit différente d'une exécution à l'autre.
- c) Le tri shaker repose sur la même technique que le tri à bulle, à la différence que, une fois sur deux, le vecteur est examiné de gauche à droite, tandis que l'autre fois sur deux il est examiné de droite à gauche. Ainsi, alternativement, les nombres de « plus grands éléments » en place et de « plus petits éléments » en place vont augmenter.
Ecrire une fonction `tri_shaker`, en s'inspirant de `tri_bulle1`, et la tester.