Université de Marne la Vallée
Licence EEA
Année 2001-2002

# Langage C

TD N° 7 : Histoire d'un pointeur qui pointe sur un pointeur qui ...

**Exercice 1**
Ecrire un programme de multiplication de deux matrices. Les dimensions des matrices seront tout d'abord précisées par l'utilisateur et les matrices seront allouées dynamiquement, puis saisies.

**Exercice 2 (Examen 1999-2000)**
Le but de cet exercice est de calculer des statistiques élémentaires sur un texte fourni par l'utilisateur. Ce texte pourra être saisi au clavier ou lu dans un fichier à l'aide d'une redirection de l'entrée standard. Dans ce cas, on tapera :
`sujet3 < nom_fichier` si `sujet3` est le nom donné à l'exécutable.
Les informations extraites du texte seront stockées sous la forme d'une structure :

```
struct texteinfo {
        unsigned long nbligne,nbspace,nbmaj,nbmin,nbcar;};
```

où `nbligne`, `nbspace`, `nbmaj`, `nbmin`,`nbcar` correspondent respectivement aux nombres de lignes, espaces, majuscules, minuscules et caractères présents dans le texte considéré.

1. Ecrire une fonction réalisant simultanément la saisie du texte et le calcul des cinq paramètres définis ci-dessus. La fonction renverra une structure de type `texteinfo`.
2. Ecrire une fonction dont l'argument sera un pointeur sur cette structure et qui affichera les résultats de la recherche précédente. On indiquera aussi en terme de pourcentage du nombre total de caractères, la quantité d'espaces, majuscules et minuscules.
3. Mettre en œuvre les deux fonctionnalités au sein du même programme.

**Exercice 3 (Examen 2000-2001)**
Le but de cet exercice est de réaliser un programme de calcul de l'intégrale d'une fonction réelle $x(t)$ (qu'on supposera intégrable au sens de Riemann) sur un intervalle $[a, b]$ (avec $b > a$).

1. Une méthode simple consiste à diviser $[a, b]$ en $N \in \mathbb{N}^*$ intervalles (de longueurs suffisamment petites) et à approximer l'intégrale par
$$\sum_{n=1}^{N} I(t_{n-1}, t_n)$$
où $a = t_0 < \cdots < t_N = b$ et
$$I(t_{n-1}, t_n) = (t_n - t_{n-1}) \frac{x(t_{n-1}) + x(t_n)}{2} \ .$$
Ecrire une fonction C `integ(float *t,int N)` permettant de calculer l'intégrale de $x(t) = t^2$. Les arguments de cette fonction seront :
– un pointeur `t` sur un tableau de flottants définissant la suite $(t_n)_{0 \le n \le N}$
– le nombre `N` d'intervalles.
Mettre en œuvre la méthode d'intégration quand les intervalles sont de longueur égale à $(b - a)/N$.

2. On a intérêt à subdviser les intervalles où la fonction varie fortement de façon à affiner l'approximation précédente. Pour cela, on part de $N$ intervalles de même longueur et on procède de la manière suivante : on se fixe un seuil $s > 0$ et, pour chaque $n \in \{1, \ldots, N\}$, si $|x\big(\frac{t_{n-1}+t_n}{2}\big) - x(t_{n-1})| > s$ ou si $|x(t_n) - x\big(\frac{t_{n-1}+t_n}{2}\big)| > s$, on découpe en deux l'intervalle $[t_{n-1}, t_n]$, ce qui équivaut à ajouter comme point intermédiaire $(t_{n-1} + t_n)/2$ entre $t_{n-1}$ et $t_n$. Ecrire une fonction C `subdvi(float *t,int N,float s)` effectuant la subvision que nous venons de décrire. La fonction renverra une structure

   ```
   struct resultat {float *tt; int NN};
   ```

   permettant de récupérer les caractéristiques de la suite d'intervalles produite. A partir de cette fonction, réaliser une fonction d'intégration améliorée `integam(float *t,int N,float s)`.

3. Modifier les fonctions précédentes de façon à pouvoir prendre en compte une fonction $x(t)$ quelconque. On fera appel à un pointeur de fonction qui sera passé en argument.