Semestre 1, Licence MASS Informatique – TD 6 (Lundi 14 à vendredi 18 novembre 2005)
(pas de TD 5)

Objectifs de la séance:

1. encore boucles et tests
2. 1er contact avec la bibliothèque standard <math.h>
3. Une application du schéma de Horner

Exercice 1

Ecrire un programme qui demande à un commercial son chiffre d'affaire, le lit et calcule le montant de sa prime de rendement.

La prime de rendement est égale à 1% de la partie du chiffre d'affaire comprise entre 20 000 euros et 50 000 euros, plus 2% de ce qui dépasse 50 000 euros.

Le résultat devra être affiché d'une des façons suivantes :

vous ne touchez aucune prime

vous touchez une prime de <*montant de la prime*> euros.

Exercice 2

Ecrire un programme qui calcule la somme:

$$1 + x + x^2/2! + x^3/3! + ... + x^n/n!$$

Comparer avec le résultat de la fonction exp de <math.h> de prototype *double exp(double x)*.

Pour utiliser une telle fonction dans un programme il faut:

a) placer la ligne
#include <math.h>
en tête du programme.

b) utiliser l'option -lm dans la commande de compilation gcc. par exemple:

gcc -lm -o monprog monprog.c	pour créer l'exécutable ./monprog
gcc -lm monprog.c	pour créer l'exécutable ./a.out

Exercice 3 : schéma de Horner.

(cas particulier de la conversion d'un nombre dans la base de calcul)

Ecrire un programme qui lit un nombre binaire au clavier, "calcule" sa valeur dans une variable entière, puis l'affiche dans notre représentation décimale habituelle.

Pour cela on lira les chiffres binaires un à un, de gauche à droite, et on calculera au fur et à mesure la valeur de l'entier correspondant.

Soit le nombre binaire $c_p\ c_{p-1}...c_1c_0$, sa valeur est $n = ((...(c_p.2+c_{p-1}).2+ ...).2 + c_1).2 + c_0$

Ceci peut se traduire, mathématiquement, par une suite de valeurs:

$u_0 = c_p$

$u_i = u_{i-1}.2 + c_{p-i}$ pour tout i compris entre 1 et p

Aide: Pour lire les chiffres un à un il vaut mieux lire les chiffres en tant que caractères avec la fonction getchar (dans une instruction telle que c=getchar(); par exemple) et transformer le chiffre (si c'est bien un chiffre) en son équivalent numérique par une instruction telle que i = c - '0'; .