DEUG MASS Info1 – TD 7 (Lundi 21 à vendredi 25 novembre 2005)

Objectifs de la séance:

1. encore des boucles et tests, sommation de suites mathématiques.
2. caractères et nombres entiers en C.
3. définition et utilisation d'une fonction.

Exercice 1: Suite de Fibonacci.

Cette suite est définie par récurrence: $u_0 = u_1 = 1$

$u_i = u_{i-1} + u_{i-2}$ pour $i>1$

Ecrire un programme qui demande la valeur de l'entier n à l'utilisateur puis qui calcule et affiche la valeur de l'élément u_n de la suite de Fibonacci.

Exercice 2: un peu de dessin.

Ecrire un programme qui lit la valeur de l'entier n et affiche un triangle isocèle de n lignes de hauteur et dont la base est constituée de 2n-1 étoiles.

Pour n=4, par exemple ce la donnera :

```
   *
  ***
 *****
*******
```

Exercice 3: manipulation (très) élémentaire de caractères.

Ecrire un programme qui demande un caractère à l'utilisateur et l'affiche ainsi que son code ASCII en décimal (spécification %d de printf), en octal (%o) et en hexadécimal (%X).

Exercice 4: Ecrire un programme qui lit la valeur de l'entier n, puis qui calcul la somme des n premiers entiers, la sommes de leurs carrés, la somme de leurs puissances 3 et la somme de leurs puissances 4. On calculera aussi ces sommes à l'aide des formules $S1=n.(n+1)/2$, $S2=n.(n+1).(2n+1)/6$, $S3=n^2.(n+1)^2/4$, $S4=n.(n+1).(2n+1).(3n^2+3n-1)/30$. En dehors du TD on pourra s'entraîner à démontrer ces formules, soit par récurrence comme déjà vu pour S2 en mathématique, soit autrement.

Exercice 5: définition et utilisation d'une fonction.

5.1. Ecrire une fonction d'entête *int sommediv(int n)* qui donne la somme des diviseurs de la valeur transmise dans n. On considérera les seuls diviseurs inférieurs strictement à n; ainsi sommediv(10) devra valoir 8 (soit 5+2+1).

5.2. Ecrire un programme uniquement pour tester la fonction sommediv.

5.3. On dit que 2 nombres sont amis si la somme des diviseurs de l'un (comme en 5.1) est égal à l'autre et réciproquement. On pourra facilement vérifier que 220 et 284 sont amis. Paradoxalement, on dit qu'un nombre est parfait, s'il est ami avec lui même, autrement dit s'il est égal à la somme de ses diviseurs.

5.3a. En n'utilisant que la fonction sommediv, donner une méthode simple et efficace pour dire si un nombre p est parfait, s'il a un nombre ami et le quel.

5.3b. Ecrire un programme qui donne tous les nombres parfaits p, les paires de nombres amis (a,b) tels que $p < N$, $a < b$, $a < N$ où N est une constante qu'on définira (à 1000 par exemple). Bien sûr le programme devra utiliser la fonction sommediv définie en 5.1, et ne comporter qu'une seule boucle.