INF 111, L1, 2004/2005

Feuille 3bis

30 Septembre 2004

For/While, exos (bis)

Laure Gonnord `http://www-verimag.imag.fr/~gonnord/`

1 Avec une(des) boucle(s) for

On écrira les FONCTIONS suivantes :

1. Une fonction `puissance` qui prend en arguments deux entiers x et k et qui retourne x^k. La fonction aura donc l'en-tête suivante : `int puissance(int x,int k)` On pourra utiliser une variable auxiliaire `tmp` qui contient successivement x^2, puis x^3, puis x^4, etc.
2. Soit la suite u suivante : $u_n = \begin{cases} A & \text{si } n = 0, \\ 2 + 3 * u_{n-1} & \text{sinon.} \end{cases}$.

   Écrire une fonction d'en-tête `int donne_suivant(int u)` qui étant donné un terme de la suite, retourne le suivant (c'est-à-dire, que si on lui donne u_{15}, donne u_{16} = ...). Utiliser cette fonction dans le corps de la fonction `main` pour imprimer les 10 premiers termes de la suite pour $A = 3$.
3. Soit la suite F suivante : $F_0 = 0$, $F_1 = 1$ et pour tout $n \in \mathbb{N}$, $F_{n+2} = F_{n+1} + F_n$. Montrer comment calculer à la main F_{10}. Écrire une fonction `int donne_suivant(int a,int b)` qui étant donnés 2 termes successifs de la suite donne le suivant. Utiliser cette fonction dans le corps de la fonction `main` pour imprimer les 10 premiers termes de la suite (bien remarquer que l'on a cette fois-ci besoin d'au moins 2 variables, en plus de l'indice courant).
4. Écrire une fonction `int sommation(int n)` qui étant donné un entier $n > 0$ renvoie $\sum_{i=1}^{n} i$ (la somme des n premiers entiers).
5. Écrire une fonction qui étant donné un entier $n > 0$, renvoie $\sum_{i=1}^{n} \big(\sum_{j=1}^{i} (i + j) \big)$.

   (On utilisera 2 boucles for imbriquées, l'une avec i comme indice de boucle, l'autre avec j.)

2 Boucle while

1. On reprend la suite u précédente (section 1, exo 2). Écrire un programme qui affiche les entiers de la suite tant qu'ils sont plus petits qu'un entier donné par l'utilisateur. On pourra utiliser la fonction `donne_suivant`. On affichera aussi la valeur d'indice du dernier nombre imprimé (utiliser une variable auxiliaire qui compte le nombre de tours de boucles).
2. Écrire un programme qui demande une borne supérieure à l'utilisateur et imprime les entiers qui sont à la fois multiples de 3 et multiples de 5 et qui sont inférieurs à cette borne. On demande d'utiliser uniquement une boucle while.
3. Écrire un programme qui calcule et affiche le plus petit entier n tel que $\sum_{i=1}^{n} \big(\sum_{j=1}^{i} (i + j) \big) \geq 1000$.
4. Écrire un PROGRAMME `bilan` qui lit une suite de nombres positifs (stricts) ou négatifs (stricts) terminée par 0 et qui calcule et affiche la somme des recettes (les >0), la somme des dépenses (les <0) et le bilan (la différence entre les recettes et les dépenses). On s'attachera à rendre le programme lisible, éventuellement en utilisant des fonctions auxiliaires.