Département IMA / 3A (S5) **Programmation Structurée 2012/2013**
http://laure.gonnord.org/pro/teaching/

TD2 - Algorithmique et C de base - Boucle WHILE

Objectifs

Conception de programmes simples, utilisation des boucles «tant que». Premières fonctions et actions.

1 Programmes C

On écrira directement des programmes C complets.

Exercice 1 *Soit la suite u suivante :* $u_n = \begin{cases} A & \textit{si } n = 0, \\ 2 + 3 * u_{n-1} & \textit{sinon.} \end{cases}$.
Écrire un programme complet qui imprime les 10 premiers termes de la suite pour $A = 3$.

Exercice 2 *Écrire un programme C complet qui demande un entier* $n > 0$ *à l'utilisateur, calcule et imprime* $\sum_{i=1}^{n} i$.

2 Algorithmique

On ne traduira pas les algorithmes en langage C.

Exercice 3 *On suppose qu'on dispose d'une fonction* `nb_au_hasard` *de signature Entier × Entiers → Entier. L'appel* `nb_au_hasard(nmin,nmax)` *retourne un entier tiré au hasard entre* `nmin` *et* `nmax` *compris.*

Concevoir un programme *qui :*
- *Choisit au hasard un nombre x compris entre 0 et 64.*
- *Demande à l'utilisateur un nombre y, dit "gagné" si* $x = y$ *, et dans le cas contraire donne une indication "plus petit" ou "plus grand".*
- *Lorsque le nombre a été deviné, imprime le nombre de tentatives.*

A partir de maintenant, nos algorithmes seront des fonctions ou des actions

Exercice 4 *Soit* u_n *définie par* $u_0 = 1515$ *et* $u_{n+1} = 3u_n + 42$ *pour* $n \in \mathbb{N}^*$.
- *Écrire une fonction impérative* `calculeU` *qui étant donné k, calcule* u_k.
- *Écrire un algorithme qui calcule le premier k tel que* u_k *est strictement supérieur à 1 million, en utilisant ou pas des appels à la fonction précédente.*

Exercice 5 (Calcul de valeur approchée de Pi par somme de série)

"On sait que" $\frac{\pi}{4} = arctan(1) = \displaystyle\sum_{n=0}^{\infty} \frac{(-1)^n}{2*n+1}$.

Soit ε un réél fixé petit (10^{-5} par exemple). Écrire une fonction qui calcule une valeur approchée de π à ε près. On calculera des termes successifs et on s'arrêtera lorsque deux termes successifs auront une différence inférieure à ε.

Ce genre d'algorithme sera courant en analyse numérique au S6.

Exercice 6 *Concevoir un algorithme qui calcule et affiche le plus petit entier n tel que* $\displaystyle\sum_{i=1}^{n} \Big(\sum_{j=1}^{i} (i+j) \Big) \geq 1000$.