Université Blaise Pascal

Licence 1ère année
Année 2009-2010

Algorithmique et programmation en C
(11INFF2)

Feuille 3 : boucles *while*

Exercice 1 (TD+TP)**.** Écrivez un algorithme (resp. programme (en TP)) qui lit une suite d'entiers terminée par -1 et qui affiche la moyenne de ces nombres.

Exercice 2 (TD+TP)**.** Écrivez un algorithme qui détermine si une suite d'entiers terminée par -1 est croissante ou pas.

Exercice 3 (TD+TP)**.** Écrivez un algorithme qui détermine si une suite d'entiers terminée par -1 est croissante, décroissante, constante ou aucun des trois.

Exercice 4 (TD)**.** Écrivez un algorithme qui calcule le plus petit nombre i, supérieur à 100, tel que $3i + 1$ est multiple de 11.

Exercice 5 (TD)**.** Écrivez un algorithme qui calcule le plus petit nombre i, supérieur à 100, tel que i est multiple de 7 et $2i - 3$ est multiple de 5.

Exercice 6 (TD)**.** Écrivez un algorithme qui calcule le plus grand n tel que :

$$\sum_{i=0}^{n}(2i + 3) < 100.$$

Exercice 7 (TD+TP)**.** Le but de cet exercice est de deviner un nombre que l'ordinateur aura choisi. Pour que le programme puisse choisir un nombre aléatoire (c'est-à dire que le nombre change à chaque fois que le programme est exécuté), il faut :

- inclure les bibliothèques `stdlib.h` et `time.h`,
- appeler `srand(time(NULL)) ;` en début de programme,
- récupérer le nombre aléatoire n en faisant `n = rand()%100+1 ;`.

L'utilisatrice propose des nombres un par un, et l'ordinateur indique à chaque fois si c'est le nombre recherché, si le nombre proposé est strictement plus petit que celui recherché, ou si le nombre proposé est strictement plus grand que le nombre recherché. Le programme affiche le score, c'est-à-dire le nombre de tentatives.

Exercice 8 (TD+TP)**.** À présent, c'est à l'utilisatrice de penser à un nombre et à l'ordinateur de le deviner. L'utilisatrice indiquera avec 0 que le nombre proposé par l'ordinateur est le nombre auquel elle pensait, avec -1 que le nombre proposé par l'ordinateur est trop petit, et avec 1 que le nombre proposé par l'ordinateur est trop grand. Une fois le nombre trouvé, le programme affichera son propre score (qui est le nombre de propositions faites).

Exercice 9 (TD)**.** Quelle est la meilleure stratégie pour gagner au jeu précédent ? En combien de propositions est-on sur de pouvoir gagner pour des nombres choisis entre 1 et 100 ? Pour des nombres choisis entre 1 et 10000 ? Pour des nombres choisis entre 10001 et 20000 ?

Exercice 10 (TD)**.** On a vu dans le TD précédent que l'on pouvait approcher $\sqrt{x}$ en utilisant la suite u_n définie par $u_0 = x$ et $u_{n+1} = (u_n + x/u_n)/2$. Calculez une approximation de $\sqrt{x}$ à ε près. On peut remarquer que si $u_i - x/u_i > \varepsilon$, l'approximation u_i n'est pas assez bonne.

Exercice 11 (TD)**.** Soit f la fonction de Syracuse (vue en cours) définie par :

- $f(1) = 1$,
- $f(n) = n/2$ pour tout n pair,
- $f(n) = 3n + 1$ pour tout n impair et différent de 1.

Selon la conjecture de Syracuse, pour tout entier n, il existe m tel que $f^m(n) = 1$. En d'autres termes, en appliquant m fois la fonction f à partir de n, on obtient 1. Par exemple, pour $n = 3$ on a $m = 7$. En effet : $f(3) = 10$, $f^2(3) = f(10) = 5$, $f^3(3) = f(5) = 16$, $f^4(3) = 8$, $f^5(3) = 4$, $f^6(3) = 2$ et $f^7(3) = 1$. Écrivez un algorithme qui retourne la valeur de m pour un entier n donné en paramètre. Il n'est pas demandé de faire une fonction.

Exercice 12 (TD)**.** Écrivez un algorithme qui affiche le premier entier n pour lequel il faut appliquer la fonction f plus de 10 fois pour obtenir 1.

Exercice 13 (TD+TP)**.** Comme la conjecture de Syracuse n'est pas prouvée, il existe peut-être des entiers n pour lesquels on n'arrive jamais à 1 en appliquant la fonction f. Pour ne pas être bloqué, on choisit d'appliquer la fonction f au plus max fois. Écrivez un programme qui demande à l'utilisatrice un entier n et un entier max et qui applique la fonction f jusqu'à obtenir 1, au plus max fois. Le programme devra afficher si la valeur 1 a été obtenue ou si le nombre maximum d'applications a été atteint.