TP3 : Les boucles indéfinies

Guillaume Aubian

Ce document a été grandement inspiré par – et largement copié sur – le document correspondant du responsable précédent de ce cours, Juliusz Chroboczek, avec son accord.

1 Merci c'est gentil

- Écrivez un programme qui affiche « J'adore ce TP ! » indéfiniment (jusqu'à ce qu'il n'y ait plus de courant).
- Un appel à `time(NULL)` retourne l'heure courante, mesurée en secondes depuis le premier janvier 1970. Modifiez votre programme pour qu'il affiche « J'adore ce TP ! » pendant 10 secondes puis termine. Que se passe-t-il si on ajuste l'horloge de l'ordinateur pendant qu'il s'exécute ?

2 Un de plus s'il-vous-plaît

- Écrivez un programme qui lit des entiers au clavier jusqu'à ce que l'utilisateur tape un 0, puis affiche leur somme.
- Modifiez votre programme pour qu'il affiche la somme et le produit des entiers rentrés. (Attention, la liste est toujours terminée par 0, pas par 1.)

3 Combien de chiffres ?

Écrivez un programme qui lit un entier entré par l'utilisateur et affiche le nombre de chiffres de cet entier en représentation canonique en base 10 (la représentation usuelle, sans zéros au début). Votre programme ne devra pas utiliser d'autres types que int.

4 Syracuse

Pour chaque entier m, on définit la m-ième suite de Syracuse $(S^{(m)})$ par

$$S_0^{(m)} = m$$

$$S_n^{(m)} = S_{n-1}^{(m)}/2 \text{ si } S_{n-1}^{(m)} \text{ est pair}$$

$$S_n^{(m)} = 3S_{n-1}^{(m)} + 1 \text{ sinon.}$$

- Écrivez (à la main) les premiers termes de $(S^{(6)})$ jusqu'à atteindre la valeur 1. Même question pour $(S^{(11)})$. (Ne le faites pas pour $(S^{(27)})$.)
- La Conjecture de Collatz dit que pour tout entier m, $(S^{(m)})$ atteint 1. On appelle temps de vol de m le plus petit entier n tel que $S^(m)_n$ vaut 1. La Conjecture de Collatz dit donc que le temps de vol de tout entier m est fini. Écrivez un programme qui lit un entier m puis affiche le temps de vol de m. Pourquoi vous ai-je demandé de ne pas calculer $S^{(27)}$?
- Écrivez un programme qui vérifie la Conjecture de Collatz pour les entiers compris entre 1 et 100.

5 Le juste prix

Écrivez un programme qui choisit aléatoirement[1] un nombre s compris entre 1 et 100 (au sens large), et demande à l'utilisateur de le deviner. À chaque fois que l'utilisateur entre un nombre n,

- si $n < s$, votre programme affichera « Trop petit ! » et recommencera ;
- si $n > s$, votre programme affichera « Trop grand ! » et recommencera ; enfin,
- si $n = s$, votre programme affichera « Gagné ! » suivi du nombre de coups que l'utilisateur a mis pour deviner le nombre, et terminera.

6 Prix juste le

Écrivez un programme qui devine un nombre entre 1 et 100 choisi par l'utilisateur. Votre programme utilisera 3 valeurs `p` (petit), `m` (moyen) et `g` (grand). À chaque étape, votre programme devinera l'entier `m` et demandera à l'utilisateur s'il est juste ; l'utilisateur devra entrer -1 (trop petit), 0 (juste) ou $+1$ (trop grand). Initialement, `p` vaudra 1 et `g` vaudra 101. À chaque étape, votre programme fera $m = \lfloor \frac{p+g}{2} \rfloor$, et devinera m. Si m est juste, votre programme a gagné, et il termine. Si m est trop petit, votre programme fera $p = m + 1$; sinon, $g = m - 1$. Votre programme terminera en râlant lorsque $p = g$.

7 Le second premier

Écrivez une version plus efficace du programme `premier.c` du TP précédent qui arrête l'itération dès lors qu'un diviseur a été trouvé.

[1] Faites d'abord `srand(time(NULL));` pour initialiser le générateur de nombres pseudo-aléatoires, puis `rand()` pour générer un entier pseudo-aléatoire.

8 Une certaine harmonie

La moyenne harmonique H d'une famille de nombres $(a_i)_{1 \leq i \leq n}$ est définie par $H = 0$ s'il existe k tel que $a_k = 0$ et

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{a_i}}$$

sinon.

Écrivez un programme qui lit un entier n, puis qui lit n nombres à virgule flottante et affiche leur moyenne harmonique. (Il faudra lire n nombres même si l'un d'entre eux est nul.)