Initiation à la Programmation en C (L1 CPEI)*

TP 2 : Branchements et boucles

30/01/2019

Exercice 1 (La suite de Syracuse). À partir d'un certain entier de départ a, on définit la *suite de Syracuse* comme suit:

$$\begin{aligned} u_0 &= a, \\ u_{n+1} &= \frac{u_n}{2} && \text{si } u_n \text{ est pair,} \\ u_{n+1} &= 3u_n + 1 && \text{si } u_n \text{ est impair.} \end{aligned}$$

Une célèbre conjecture[a] affirme que cette suite atteint toujours la valeur 1 quelle que soit sa valeur de départ.

(1) Écrivez un programme qui demande un entier a à l'utilisateur et qui affiche le premier indice n pour lequel $u_n = 1$.

(2) Modifiez votre programme pour qu'il demande un deuxième entier b à l'utilisateur et affiche le premier indice n pour lequel $u_n = b$ ou `"Nope !"` si b n'apparaît pas dans la suite. Remarquez qu'après avoir atteint 1, la séquence suivante se répète à l'infini: $1, 4, 2, 1, 4, 2, \ldots$.

[a] Encore non résolue à ce jour !

*Cours donné par prof. Roberto Amadio. Moniteur 2019 : Cédric Ho Thanh. TPs/TDs basés sur ceux des précédents moniteurs : Florien Bourse (2017), Antoine Dallon (2018). Autres contributeurs : Juliusz Chroboczek, Gabriel Radanne.

Exercice 2 (Le nombre d'or)**.** Nous rappelons ici la méthode de Newton pour calculer un antécédent de 0. Le problème à résoudre est le suivant : étant donné une fonction f, on cherche x tel que $f(x) = 0$. Pour $(x_i)_{i\in\mathbb{N}}$ une suite tendant vers x, on a:

$$f'(x_i) \simeq \frac{f(x_i)}{x_i - x_{i+1}} \qquad \text{d'où} \qquad x_{i+1} \simeq x_i - \frac{f(x_i)}{f'(x_i)}.$$

On utilisera la condition d'arrêt $|x_{i+1} - x_i| < \epsilon$ pour un ϵ petit, correspondant au degré d'approximation que l'on souhaite.

Le nombre d'or $\varphi = \frac{1+\sqrt{5}}{2}$ est l'une des solutions de l'équation $x^2 = x + 1$.

(1) Quel est un bon choix de fonction s'annulant en φ ? Calculer sa dérivée.

(2) Sachant que φ est proche de 1.6, utilisez la méthode de Newton pour en calculer une approximation à 20 décimales.

Exercice 3 (Devine mon âge !)**.** Vous voulez créer un programme qui devine un nombre entre 1 et 100 auquel l'utilisateur pense. Le programme devra faire des propositions auxquelles l'utilisateur pourra répondre “c'est plus” en tapant `1`, “c'est moins” en tapant `-1`, ou “tout pile” en tapant `0`.

Pour deviner le nombre, le programme va procéder par *dichotomie*: il garde en mémoire une borne inférieure *min* et une borne supérieure *max* (initialisées à 1 et 100 respectivement) sur le nombre à deviner, et à chaque étape va diviser le nombre de réponses possibles par 2 en demandant si le nombre est plus petit ou plus grand que $min + \frac{max-min}{2}$.

Attention ! Le programme s'arrête quand $max = min + 1$ et teste *min* et *max*. En effet, à cet étape, on peut avoir $\frac{max-min}{2} = 0$ (division entière) et attendre $min = max$ pourrait causer des boucles infinies...

Le programme devra terminer quand l'utilisateur tape 0 en écrivant `"Bien joué !"`, ou après avoir épuisé tous les nombres possibles entre 1 et 100 en écrivant `"Tricheur !"`.

Exercice 4 (Petits dessins)**.** (1) Écrivez un programme qui affiche avec des étoiles, en demandant sa hauteur à l'utilisateur. Par exemple, si la hauteur demandée est 5, le programme devra afficher :

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

(2) Écrivez un programme qui affiche un triangle renversé, en demandant sa hauteur à l'utilisateur. Voici un triangle renversé de hauteur 5 :

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

(3) Écrivez un programme qui affiche un sapin, en demandant la hauteur du feuillage à l'utilisateur. La base du tronc aura toujours la même forme. Le programme refusera de dessiner un sapin de hauteur inférieure à 3. Voici un sapin dont le feuillage a une hauteur de 5 :

```
    *
   ***
  *****
 *******
*********
   |||
```