

Programmation en C et structures de données

guillaume.revy@univ-perp.fr

Structures itératives

Exercice 1. Quelques exercices sur les boucles

- 1. Écrire un programme qui lit une valeur entière n au clavier, puis calcule et affiche la factorielle de n . Proposer plusieurs versions, utilisant différents types de boucles.
- 2. Écrire un programme qui lit un entier n au clavier, puis qui détermine et affiche le nombre de chiffres non nuls dans son écriture décimale.

Exercice 2. Nombre premier vs. nombre parfait

Pour rappel, un nombre premier est un nombre qui n'est divisible uniquement par 1 et par lui-même (ainsi 1 n'est pas premier).

- 1. Écrire un programme qui lit un nombre entier n au clavier, puis teste et affiche si ce nombre est premier ou pas.
- 2. Étendre ce programme pour afficher tous les nombres premiers compris entre 1 et k , où k est un entier lu au clavier.

Maintenant, un nombre parfait, quant à lui, est un nombre n qui est égal à la somme de ses diviseurs stricts (c'est-à-dire, n exclu).

- 3. Écrire un programme qui lit un nombre entier n au clavier, puis qui détermine si ce nombre est parfait.
- 4. Modifier le programme pour afficher tous les nombres parfaits dans l'intervalle $[1, k]$.

Exercice 3. Moyenne d'une liste d'entiers

- 1. Écrire un programme qui permet de saisir une série d'entiers, et de calculer la moyenne des valeurs saisies. Terminer la saisie par CTRL-D. La syntaxe suivante pourra être utilisée.

```
int value;
if (scanf("%d", &value) != EOF) {    // lit la valeur de value au clavier
    ...                             // -> si CTRL-D : scanf retourne EOF
}
```

Exercice 4. Minimum et maximum d'une série d'entiers

- 1. Écrire un programme qui permet de saisir une série d'entiers, et de déterminer les valeurs minimum et maximum saisies.

Exercice 5. Comment trouver une valeur tirée aléatoire ?

On souhaite écrire un programme qui tire une valeur aléatoire entre 0 et n , que l'utilisateur devra trouver en un nombre de coup donné n_c . La syntaxe C suivante permet de tirer une valeur aléatoire.

```
srand(time(NULL));                // initialise la "graine" pour le generateur
                                  // pseudo-aléatoire
int value = rand();               // tire une valeur pseudo-aléatoire en 0 et RAND_MAX
                                  // RAND_MAX = 2147483647 (defini dans stdlib.h)
```

- 1. Écrire un programme qui tire une valeur aléatoire pour $n = 100$, par exemple. Ensuite, écrire une boucle qui demande à l'utilisateur de saisir une valeur, tant que cette valeur est différente de celle tirée aléatoirement. À chaque saisie, indiquer si la valeur à trouver est supérieure, inférieure ou égale à celle saisie. Au bout de $n_c = 10$, par exemple, si l'utilisateur n'a pas trouvée cette valeur, afficher un message lui indiquant qu'il a perdu.

Exercice 6. Multiplication égyptienne

Soient deux entiers a et b , et on souhaite calculer $r = a \times b$. On initialise r à 0. Ensuite la multiplication égyptienne fonctionne de la manière suivante :

1. si b est égal à 0, on renvoie r ,
 2. si b est pair, on multiplie a par 2 et on divise b par 2,
 3. sinon, on enlève 1 à b et on ajoute a à r ,
 4. et on recommence l'étape 1.
- 1. Écrire un programme itératif qui lit deux entiers a et b et calcule puis affiche le produit $a \times b$.

Exercice 7. Les valeurs d'une suite

On considère la suite définie de la manière suivante :

$$\begin{cases} u_0 &= a \text{ avec } a \in \mathbb{N}, \\ u_{n+1} &= u_n/2 \text{ si } u_n \text{ est pair,} \\ u_{n+1} &= 3 \times u_n + 1 \text{ sinon.} \end{cases}$$

Pour toutes les valeurs a , il existe un entier k tel que $u_k = 1$ (conjecture admise).

- 1. Écrire un programme qui lit deux valeurs entières a et k au clavier, et qui affiche toutes les valeurs de u_n de $n = 1$ à $n = k$.
- 2. Écrire un programme qui lit une valeur entière ℓ au clavier, puis qui cherche la valeur de a comprise entre 2 et ℓ qui maximise la valeur de k où $u_k = 1$. Le programme affichera cette valeur et celle de k .

Exercice 8. C'est bientôt Noël!

- 1. Écrire une programme qui, étant donné un nombre ℓ de lignes, affiche un sapin de Noël, comme celui ci-dessous (à gauche) pour $\ell = 9$.

Hauteur du sapin = 9

```

      ^
     ^ ^
    ^ ^ ^
   ^ ^ ^ ^
  ^ ^ ^ ^ ^
 ^ ^ ^ ^ ^ ^
^ ^ ^ ^ ^ ^ ^
^ ^ ^ ^ ^ ^ ^ ^
^ ^ ^ ^ ^ ^ ^ ^ ^
^ ^ ^ ^ ^ ^ ^ ^ ^ ^

```

Hauteur du sapin = 9
 Hauteur du tronc = 2
 Rayon du tronc = 3
 Probabilite = 20

```

      ^
     ^ ^
    ^ ^ ^
   o ^ o o o ^ ^
  o o ^ ^ ^ o ^ ^
 ^ ^ ^ ^ ^ ^ ^ ^
 ^ ^ o ^ ^ ^ ^ o ^ o ^
 ^ ^ ^ ^ ^ ^ ^ o ^ o ^ ^
 ^ ^ ^ o o o o ^ ^ ^ ^ ^
      + + + + +
      + + + + +

```

- 2. Compléter le programme, pour dessiner le tronc (centré) de hauteur h et rayon r .
- 3. Compléter à nouveau le programme, pour ajouter des boules de Noël de manière aléatoire (comme ci-dessus, à droite) avec une probabilité de $p\%$ ($p \in \mathbb{N}$ lue au clavier). Pour cela, utiliser la fonction `rand`.