# TP 5bis : boucles for, tableaux (suite)

Informatique Fondamentale (IF121)
24–28 novembre 2003

## 1 Figures

Le but de ces exercices est de « dessiner » quelques figures simples en mode texte. Ces figures sont en fait composées de caractères (signes `-`, `|`, `+`, `*`, etc.). On les affiche ligne par ligne.

**Exercice 1 :** *Préliminaire*
Écrire un méthode `repètePrint` qui prend comme argument un entier $n$ et un caractère, et qui affiche $n$ fois de suite (sans passer à la ligne) le caractère en question.

**Exercice 2 :** *Rectangle plein*
Écrire un programme qui affiche un rectangle de « `*` ». Le programme commencera par demander la hauteur et la largeur du rectangle à l'utilisateur. Par exemple :

```
> java RectanglePlein
Hauteur ? 4
Largeur ? 10
**********
**********
**********
**********
```

**Exercice 3 :** *Triangle rectangle*
*(a)* Écrire un programme qui affiche un triangle rectangle isocèle composé de « * », avec l'angle droit en bas à gauche. Le programme commencera par demander la longueur du côté à l'utilisateur. Par exemple :

```
> java TriangleRectangle
Côté ? 4
*
**
***
****
```

*(b)* Même exercice, mais cette fois, on veut avoir l'angle droit en bas à droite. Par exemple :

```
> java TriangleRectangle2
Côté ? 4
   *
  **
 ***
****
```

**Exercice 4 :** *Rectangle creux*
Écrire un programme qui affiche un rectangle, en utilisant les caractères « | » et « - » pour représenter les côtés. Le programme commencera par demander la hauteur et la largeur du rectangle à l'utilisateur. Par exemple :

```
> java RectanglePlein
Hauteur ? 4
Largeur ? 10
----------
|        |
|        |
----------
```

## 2 Chaînes de caractères

Une chaîne de caractères ressemble fortement à un tableau qu'on ne peut pas modifier. Pour accéder au `i`ème caractère de la chaîne `s`, on utilise `Deug.charAt(s,i)`. La longueur de la chaîne `s` est donnée par `Deug.length(s)`.

**Exercice 5 :** *Palindrome*
Écrire un programme qui lit un mot et qui teste si ce mot est un palindrome, c'est-à-dire si on obtient le même mot en le lisant à l'envers. Par exemple, « kayak », « non », « ressasser », « selles » sont des palindromes ; « ânonna », « ressasse », « salles » n'en sont pas.

**Exercice 6 :** *Bon parenthésage*
On dit qu'un texte est bien parenthésé si et seulement si chaque parenthèse ouvrante qu'il contient est refermée par la suite. On ne tient pas compte des caractères qui ne sont pas des parenthèses. Quelques exemples :

| | |
|---|---|
| (a(b+c)) | bien |
| ()(((++))) | bien (on ne tient compte que des parenthèses) |
| (a(b+c) | mal (manque une parenthèse fermante) |
| ab+c) | mal (trop de parenthèses fermantes) |
| )( | mal (il faut ouvrir la parenthèse avant de la refermer) |

Écrire une méthode qui prend en argument une chaîne de caractères et indique si oui ou non elle est bien parenthésée.

## 3 Tableaux

**Exercice 7 :** *Retournement*
*(a)* Écrire un fragment de code qui, étant donné un tableau d'entiers `t` et deux indices `i` et `j`, échange les valeurs de `t[i]` et `t[j]`.
*(b)* Écrire un fragment de code qui retourne un tableau d'entiers `t`. Par exemple, si `t` contient au départ `{1,3,5,7,4}`, il doit contenir à la fin `{4,7,5,3,1}`.

**Exercice 8 :** *Tri par sélection*
Le but de cet exercice est d'écrire une fonction qui trie un tableau de nombres entiers.
*(a)* Écrire une fonction qui prend en argument un tableau de nombres entiers et renvoie l'indice du plus petit élément. L'expression `indiceDuMinimum(tableau)` doit renvoyer un entier `i` tel que `tableau[i]` soit le plus petit élément du tableau.
*(b)* Modifier la fonction précédente pour qu'elle prenne en argument l'indice où il faut commencer la recherche. Ainsi `indiceDuMinimumAprès(tableau,début)` doit renvoyer un entier `i` tel que `tableau[i]` est le plus petit élément du tableau d'indice supérieur ou égal à `début`.

Le principe du tri par sélection est le suivant : on commence par rechercher le plus petit élément du tableau, puis on l'échange avec le premier élément du tableau. On cherche ensuite le plus petit élément à partir du deuxième indice, puis on l'échange avec le deuxième élément du tableau, et ainsi de suite jusqu'à avoir tout trié.
*(c)* À l'aide des fonctions précédentes, écrire une fonction qui effectue un tri par sélection sur un tableau d'entiers.