

## Algorithmique T.D. n° 8

### Séquences intermédiaires et tableaux

#### Exercice 1 : Tautogramme

On considère un fichier de caractères comportant un texte formé de lettres et d'espaces. Un mot est une suite de lettres délimitée par des espaces ou par les extrémités du texte. Le texte peut commencer par des espaces. Ecrire un algorithme qui affiche un message indiquant si le texte est un tautogramme, c'est-à-dire si tous les mots du texte commencent par la même lettre.

Exemples : le lion lape le lait lentement  
 une usine urbaine utilise un ustensile unique ultrasensible  
 ce chat chaparde ces croquettes

#### Exercice 2 : Phrase palindrome

Un texte de longueur **lg** est représenté dans un tableau **t** défini sur l'intervalle  $[0..LMAX-1]$ . Spécifier et réaliser **une fonction** qui renvoie un booléen indiquant si le texte contenu dans le tableau est un palindrome, c'est-à-dire que si l'on fait abstraction des espaces, le texte se lit de gauche à droite comme de droite à gauche.

*Exemples de palindrome* : esope reste ici et se repose  
 tu l as trop ecrase cesar ce port salut  
 élu par cette crapule

#### Exercice 3 : Décalage circulaire

On considère une séquence d'entiers de longueur **lg** représentée dans un tableau **t** d'entiers défini sur l'intervalle  $[0..LMAX-1]$ ,  $0 \leq lg \leq LMAX$ . On suppose que le tableau considéré contient déjà la séquence. Spécifier et réaliser une **action nommée** qui effectue le **décalage circulaire à droite** de **t** : le 1<sup>er</sup> élément de **t** est décalé en seconde position, le second en 3<sup>ème</sup> position, l'avant dernier en dernière position, et le dernier élément se retrouve en 1<sup>ère</sup> position.

*Exemple* : Avant décalage circulaire à droite ( $lg = 6$ ) :  $t = [2, 3, 29, 54, -3, 12]$   
 Après décalage circulaire à droite ( $lg = 6$ ) :  $t = [12, 2, 3, 29, 54, -3]$

#### Exercice 4 : Matrice symétrique

Écrire une fonction qui détermine si une matrice carrée **m** d'ordre **n** est symétrique, c'est-à-dire que chaque ligne de la matrice est égale à la colonne de même indice :

$$\forall i \in [1, n], \forall j \in [1, n], m[i][j] = m[j][i]$$

fonction estSymétrique (**m** : Matrice ; **n** : entier > 0)  $\longrightarrow$  booléen  
 // estSymétrique(m,n) renvoie vrai si la matrice m d'ordre n est symétrique

Dans le lexique partagé, le type Matrice est défini comme suit :

Matrice : type tableau sur  $[1..NMAX]$  de tableaux sur  $[1..NMAX]$  de réels

## Exercice 5 : Infographie

On considère une image dont les pixels sont représentés par des caractères, représentée dans un tableau à deux dimensions composé de **nbL** lignes et de **nbC** colonnes :

**image** : tableau sur [0..nbL-1 , 0..nbC-1] de caractères

L'image est composée de faces qui doivent être dessinées dans le bon ordre. En effet, certaines faces en recouvrent d'autres et doivent donc être dessinées après ces dernières. Les faces à dessiner sont des rectangles dont les côtés sont parallèles aux bords de l'image. On connaît l'ordre dans lequel les faces rectangulaires doivent être dessinées, ainsi que la couleur de chaque face, caractérisée par un caractère particulier. Par défaut, chaque pixel est de la couleur '.'

On veut écrire un algorithme qui crée l'image à partir de la liste ordonnée des rectangles qui la composent. Les rectangles à dessiner sont représentés dans un fichier séquentiel dont chaque élément est de type **Rectangle**. Ils doivent être dessinés dans l'ordre dans lequel ils apparaissent dans le fichier. Le type **Rectangle** permet de représenter les caractéristiques d'un rectangle :

**Rectangle** : type agrégat

iLig1 : entier entre 0 et nbL-1 // indice ligne du coin supérieur gauche du rectangle  
iCol1: entier entre 0 et nbC-1 // indice colonne du coin supérieur gauche du rectangle  
iLig2 : entier entre 0 et nbL-1 // indice ligne du coin inférieur droit du rectangle  
iCol2: entier entre 0 et nbC-1 // indice colonne du coin inférieur droit du rectangle  
texture : caractère // couleur du rectangle

fagrégat

Ecrire l'algorithme qui construit l'image à partir des rectangles contenus dans le fichier, puis affiche l'image résultat à l'écran.

Exemple : on a nbL = 9 et nbC = 19

Le contenu du fichier de rectangles est le suivant (chaque ligne représente un enregistrement Rectangle) :

```
1 3 7 5 o
5 2 6 16 -
1 12 7 14 u
2 1 2 16 s
```

L'image résultat affichée par le programme est la suivante :

```
.....
...ooo.....uuu...
.sssssssssssssss..
...ooo.....uuu...
...ooo.....uuu...
..-----uuu--..
..-----uuu--..
...ooo.....uuu...
.....
```

On a d'abord dessiné le rectangle formé de caractères 'o', puis le rectangle formé de '-' qui a eu pour effet de recouvrir certains caractères 'o', puis le rectangle formé de 'u', et enfin le rectangle formé de 's', constitué d'une seule ligne.