TP 8 : tableaux a deux dimensions

Informatique Fondamentale (IF1)

Semaine du 22 novembre 2010

Préliminaires

Exercice 1. Ecrivez une fonction

```
public static int[][] saisie()
```

qui lit deux entiers `n` et `m`, puis `n` $\times$ `m` entiers, puis retourne un tableau de `n` lignes et `m` colonnes contenant les entiers saisis.

En quoi cette fonction di ere-t-elle de la fonction `Saisie` vue en cours ?

Exercice 2. Ecrivez une fonction

```
public static void affiche(int[][] a)
```

qui a che le tableau passe en parametre.

Ecrivez une fonction `main` qui saisit un tableau et l'a che ; comme d'habitude, testez votre programme.

1 Recherche de motifs

Exercice 3. Ecrivez une fonction

```
public static boolean recherche2(int[][] a, int v)
```

qui retourne `true` s'il existe une paire d'indices i, j tels que `a`[i][j] = `v`.

Ecrivez une fonction `main` qui lit au clavier un tableau ainsi qu'une valeur, et a che si la valeur se trouve dans le tableau. (Vous pouvez bien-sûr vous servir des fonctions ecrites dans la partie precedente.)

Exercice 4. Ecrivez une fonction

```
public static boolean recherche3(int[][] a, int[][] s)
```

qui retourne `true` s'il existe un sous-tableau du tableau a qui est egal a `s`. En d'autres termes, s'il existe des indices i, j tels que pour tout $k <$ `s.length` et $l <$ `s[0].length`,

`a`[i + k][j + l] = `s`[k][l].

Ecrivez une fonction `main` qui vous permette de tester votre fonction.

2 Carrés magiques

Un *carré magique* est une matrice carree de taille $n \times n$ telle que la somme de chaque rangee, de chaque colonne et de chaque diagonale ait la même valeur. Un carre magique est dit *normal* s'il contient chaque entier compris entre 1 et n^2 exactement une fois. Par exemple, le tableau suivant est un carre magique normal :

$$\begin{array}{ccccc} 2 & & & & 3 \\ & 6 & 7 & 2 & \\ 6 & & & & 7 \\ 4 & 1 & 5 & 9 & 5 \\ & 8 & 3 & 4 & \end{array}$$

Exercice 5. Ecrivez une fonction

```
public static boolean carre(int[][] a)
```

qui retourne `true` si le tableau a est une matrice carree (qui a autant de lignes que de colonnes). Ecrivez une fonction `main` et testez votre programme.

Exercice 6. Ecrivez deux fonctions

```
public static int ligne(int[][] a, int i)
public static int colonne(int[][] a, int j)
```

qui retournent la somme de la `i`-eme ligne, resp. de la `j`-eme colonne, du tableau passe en parametre. Ecrivez une fonction `main` qui vous permette de tester ces fonctions.

Exercice 7. Ecrivez deux fonctions

```
public static int diagonale1(int[][] a)
public static int diagonale2(int[][] a)
```

qui retournent la somme de la diagonale majeure, resp. de la diagonale mineure, du tableau passe en parametre. Ecrivez une fonction `main` qui vous permette de tester ces fonctions.

Exercice 8. Ecrivez un programme qui demande a l'utilisateur de rentrer un tableau, et a che s'il s'agit d'un carre magique (pas forcement normal).

Exercice 9. Ecrivez une fonction

```
public static int[] histogramme(int[][] a, int n)
```

qui retourne l'histogramme de taille n du tableau a, c'est-a-dire le tableau h de taille n tel que pour tout $i <$ n, h[i] contient le nombre d'occurrences de la valeur i dans le tableau a.

Exercice 10. Modi ez le programme ecrit ci-dessus pour qu'il a che en outre si le carre magique est normal. *(Indication : on pourra par exemple construire l'histogramme de taille* n^2 + 1 *du carré magique.)*

3 Images noir et blanc

On rappelle qu'une image noir et blanc est representee en memoire par un tableau de booleens. L'a chage d'une telle image peut se faire a l'aide de la fonction suivante :

```
public static void dessine(boolean[][] image) {
    for(int i = 0; i < image.length; i++)
        for(int j = 0; j < image[i].length; i++) {
            if(image[i][j])
                Deug.drawPoint(i, j);
        }
}
```

Exercice 11. Ecrivez une fonction

```
public static boolean[][] croix(int l, int h)
```

qui retourne une image de taille $l \times h$ contenant les segments qui connectent les milieux des bords opposes.

Ecrivez un programme ce 1a chage d'u79091 ction

Ence 11. Ecrivez une fonction

qui dessine une image en niveaux gris.

En supposant que l'image contient des zones entieres de couleur identique, comment peut-on minimiser le nombre d'appels a `setGray` ?

Exercice 15. Ecrivez une fonction

```
public static short[][] colourExpand(boolean[][] im)
```

qui convertit une image noir et blanc en une image en niveaux de gris identique. Testez votre fonction.

Exercice 16. Ecrivez une fonction

```
public static short[][] antiAlias(short[][] im)
```

qui implemente un *filtre d'anti-aliasage*. Elle retourne une nouvelle image de la même taille que l'image `im` ou les ■ creneaux ■ ont ete adoucis en utilisant les niveaux de gris. L'intensite d'un pixel de la nouvelle image sera egale a la somme de 3/4 fois la valeur du pixel correspondant de l'ancienne image et 1/4 de la moyenne des valeurs des 8 pixels voisins.

Ecrivez un programme qui a che la version ltree de la superposition des deux croix obtenues ci-dessus.