

Fiche N° 4 : Tableaux à D dimensions

Rappel d'une consigne importante pour toute la durée du module : Ne pas oublier les préconditions !!

Dans la suite on reprend les représentations définies pour les images lors du cours.

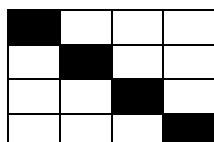
En TP, exceptionnellement, écrire les programmes « directement ». Mais l'entête doit être précisée correctement donc l'idée (principes) doit être donnée.

Premiers pas

- Écrire un algorithme qui permette d'afficher un tableau de NxP entiers (le tableau, N et P sont donnés en paramètres).
- Écrire un algorithme qui permette de déclarer et créer un tableau de NxP entiers (N et P sont donnés en paramètres) puis de l'afficher. Quelle est les 4 principales différences avec l'exercice précédent ?
- Écrire un algorithme qui permette de déclarer et créer un tableau de NxP entiers et qui l'initialise avec des 0 dans les cases (i,j) telles que i est pair, 1 pour les autres. Qu'obtenez-vous ?
- Même question mais le tableau est initialisé avec des 0 dans les cases (i,j) telles que i+j est paire 1 pour les autres. Qu'obtenez-vous ?

Exercice 1 - Images N&B

1. Écrire un algorithme qui permet de créer l'image 4 x 4 ci-dessous puis de l'afficher en remplaçant les cases noires par une * et les blanches par espace :



// En TP vous pourrez utiliser cette image pour faire vos tests d'algorithme

2. Écrire un algorithme qui affiche le nombre de pixels noirs et de pixels blancs d'une image HxL
3. Écrire un algorithme qui inverse une image HxL
4. Écrire un algorithme qui compare deux images et affiche
 - « Aucune différence » si elles sont identiques
 - « Il y a C pixels différents » où C est le nombre de cases différentes si elles ne le sont pas.
5. Écrire un algorithme qui recherche combien de fois un motif 2x2 apparaît dans une image HxL. Par exemple, le motif suivant apparaît 3 fois dans l'image de l'exercice 1.



// En TP l'appliquer à un damier 9x9

6. Même exercice mais le motif peut être de dimensions (T1 x T2) quelconques

Exercice 2 - Images Niveau de gris

1. Écrire un algorithme qui permet de créer une image telle que le niveau de gris du pixel (i,j) soit égal à $(i*255 + j*255)/(H+L)$

// En TP vous pourrez utiliser cette image pour faire vos tests d'algorithme

2. Écrire un algorithme qui binarise une image à niveaux de gris à partir d'un seuil donné par l'utilisateur : toutes les cases ayant une valeur inférieure ou égale au seuil sont mises à 0, les autres à 255.
3. Modifier l'algorithme précédent pour qu'il crée une image N&B à partir d'une image à niveaux de gris sur ce même principe de binarisation
4. Écrire un algorithme qui construit l'histogramme de l'image. Il faut construire un tableau de 256 entiers qui à la fin contient, pour chaque case l le nombre de pixels qui ont la valeur l

Exercice 3 - Images à codes couleur

On définit une image HxL à codes couleur par un tableau HxL où chaque case contient un code parmi une liste de codes ANSI prédéfinis

Couleur	Code ANSI	Valeur RGB
Noir	30	(0, 0, 0)
Rouge	31	(255, 0, 0)
Vert	32	(0, 255, 0)
Jaune	33	(255, 255, 0)
Bleu	34	(0, 0, 255)
Magenta	35	(255, 0, 255)
Cyan	36	(0, 255, 255)
Blanc	37	(255, 255, 255)
Gris foncé	90	(85, 85, 85)
Rouge clair	91	(255, 85, 85)
Vert clair	92	(85, 255, 85)
Jaune clair	93	(255, 255, 85)
Bleu clair	94	(85, 85, 255)
Magenta clair	95	(255, 85, 255)
Cyan clair	96	(85, 255, 255)
Blanc gris	97	(255, 255, 255)

Par exemple :

Soit Image : un tableau de 7x7 entiers initialisé à

```
{ {94, 94, 34, 94, 34, 34, 94},  
  {94, 34, 34, 32, 34, 94, 34},  
  {94, 34, 32, 32, 92, 34, 94},  
  {94, 34, 32, 32, 92, 34, 94},  
  {94, 92, 32, 32, 92, 34, 94},  
  {97, 97, 90, 30, 30, 97, 94},  
  {97, 97, 30, 30, 90, 97, 97}};
```

// En TP vous pourrez utiliser cette image pour faire vos tests d'algorithme

On utilisera un mécanisme qui permet d'afficher un texte en couleur directement sur une console (un écran) sans passer par des mécanismes graphiques évolués. Il suffira d'afficher des caractères par :

```
Console.Write("\u001b[32m * \u001b[0m"); // Affiche le texte `` * `` avec la couleur 32 (vert)
```

Attention : Sous WINDOWS cela peut ne pas fonctionner. Il faut alors au préalable exécuter la commande en mode Administrateur :

```
reg add HKEY_CURRENT_USER\Console /v VirtualTerminalLevel /t REG_DWORD /d 0x00000001 /f
```

Puis relancer la console

// En TP faites super attention en recopiant. Il faut absolument respecter la syntaxe !!

Dans la suite, on notera cette action par : Afficher(`` * ``, 32)

1. Écrire un algorithme qui affiche les codes contenus dans un image couleur HxL
2. Écrire un algorithme qui affiche l'image en couleur avec des caractères '*' et des espaces.

// On doit obtenir ceci :

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

Exercice 3 - Images Couleur RGB

Soit Image : un tableau de 4x4 entiers initialisé à $\{\{91, 97, 97, 30\},$
 $\{97, 92, 97, 97\},$
 $\{97, 97, 93, 97\},$
 $\{30, 97, 97, 94\}\};$

// Correspond à l'image à codes couleur suivante :

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

1. Créer une table Code qui représente la table ANSI de l'exercice 3. Expliquez comment on peut alors afficher la composition (RGB) d'un pixel (i,j) dans Image.
2. Écrire un algorithme qui affiche la composition RGK d'un pixel (i,j) donné.
(Question assez difficile) Quelle serait alors l'idée pour transformer cette image Image en image RGB ?
3. (Question difficile) Écrire un algorithme qui construit une image couleur HxL en R (rouge), G (vert) et B (bleu) à partir d'une image à codes couleur.
4. Écrire un algorithme qui construit une image à codes couleur HxL à partir d'une image en R (rouge), G (vert) et B (bleu). Préciser les préconditions