

Entraînement CCF n°4 : photographie/ négatif

Le sujet propose la manipulation d'images pixélisées.

Une image numérique matricielle (en anglais : bitmap = « carte de bits ») est essentiellement un ensemble de points (nommés pixels), organisés en un rectangle, chacun de ces points étant pourvu d'une « couleur » qui peut être un niveau de gris, une combinaison d'intensités lumineuses en rouge, vert et bleu ou simplement du noir ou blanc.

On peut comparer l'image matricielle à une feuille de papier quadrillé où l'on écrit, comme il est d'usage, de gauche à droite et de haut en bas, et où on mettrait une lettre (caractère) dans chaque case (ou pixel). La lettre inscrite dans la case permet alors d'en définir le coloriage.

On peut donc poser comme équivalence :

- un pixel noir correspond à un 1
- un pixel blanc correspond à un 0

On peut alors assimiler une image en noir et blanc à un tableau à 2 dimensions (matrice) remplis 0 et des 1.

Partie A: Algorithmique sur table

Partie A1: Voici un bout de code écrit en pseudo langage algorithmique:

Fonction mystère (n : entier)

Variables:

tab: Tableau[n][n] d'entiers

i, j, q : entiers

Début:

q ← (n-n%4)/4 # n%4 est le reste de la division de n par 4

pour i **de** 0 **à** n-1 **faire**

pour j **de** 0 **à** n-1 **faire**

si ((i < q) **ou** (i > n-1-q) **ou** (j < q) **ou** (j > n-1-q)) **alors**

 tab[i][j] ← 0

sinon

 tab[i][j] ← 1

finSi

finPour

finPour

Afficher("Voici q: ", q)

Fin

1. Que contient le tableau si n=6?
2. Que réalise la fonction mystère?

Partie A2:

Un photographe qui ne prend des photos qu'en noir et blanc, aime faire de l'effet lors du développement. Ce qu'il aime particulièrement c'est faire des négatifs. C'est-à-dire, inverser les couleurs sur l'image : les blancs sont transformés en noir et les noirs en blancs.

On appelle négatif d'une image en noir et blanc l'image dans laquelle la couleur de chaque pixel est inversée. Voici le résultat du négatif à partir de l'image de départ:

Image de départ

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |

Négatif de l'image

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |

3. Pour construire le négatif d'une image, par quelle valeur est remplacé un pixel 1? Un pixel 0?
4. Ecrivez la *fonction Négatif* (*tab[n][p]: tableau d'entiers*) reçoit en paramètre un tableau d'entiers à 2 dimensions de taille n x p (avec n et p, 2 entiers strictement positifs) contenant des 0 et des 1 et qui retourne un tableau à 2 dimensions correspondant au négatif de l'image.

Partie A3:

Notre photographe, qui trouve que la manipulation des images fascinante, aime également faire des agrandissements et plus précisément doubler la taille des images.

Pour obtenir une telle image, il considère qu'une case du tableau va prendre 4x plus de place, soit 4 cases.

5. Si l'image de départ est de taille n x p quel sera la taille de l'image une fois agrandie?
6. Ecrivez la *fonction Double* (*tab[n][p]: tableau d'entiers*) reçoit en paramètre un tableau d'entiers à 2 dimensions de taille n x p (avec n et p, 2 entiers strictement positifs) contenant des 0 et des 1 et qui retourne un tableau à 2 dimensions correspondant à la matrice de cette image deux fois plus grande

Partie B: Algorithmique sur machine

Pour cette partie, le candidat dispose d'un ordinateur équipé d'un environnement de travail pour le langage de son choix.

1. Implémenter la fonction **mystère(n)** qui reçoit en paramètre un entier n et qui retourne en sortie un tableau d'entiers à 2 dimensions n x n contenant des 0 et des 1, qui a été présenté dans la partie A1 du sujet
2. Vous implémenterez les fonction **Négatif(tab, n, p)** et **Double(tab, n, p)** qui reçoivent en paramètre un tableau de dimension n par p, n et p sont 2 entiers.
3. Vous testerez ces deux fonctions grâce à la fonction **mystère(6)**