

Problème 2 (60 min; 7 points)

Pour pouvoir commercialiser le jeu d'échec que vous aviez codé avec vos amis du cours (cfr. Examens précédents), il vous faut maintenant créer une interface graphique attrayante. Ce n'est pas une mince affaire et une fois de plus, vous vous répartissez le travail avec vos amis. Votre partie consiste à créer un filtre servant à augmenter les contrastes : La méthode `filtre` prends en entrée un tableau d'entier et renvoie un tableau correspondant à l'image filtrée. Ce filtre accentue les contrastes : l'image filtrée est bien plus nette que l'image de départ (voir image de droite du Taj-Mahal ci-dessous).

Le jeu étant (pour le moment...) en noir et blanc, ce qui doit être affiché à l'écran est encodé dans une matrice de pixels. L'intensité de chaque pixel correspond à un chiffre entier codé de 0 (pour le noir) à 255 (pour le blanc). Pour un exemple, voir ci-contre.

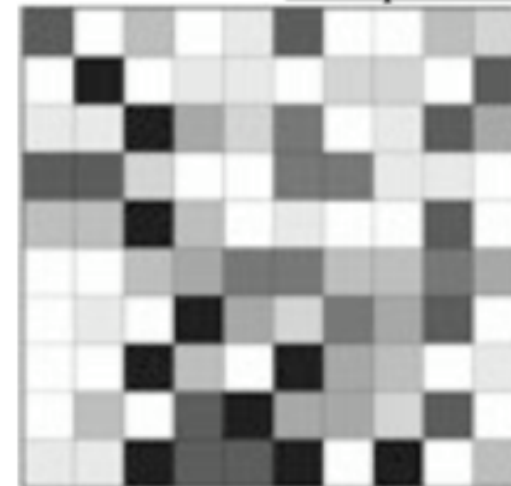
Le filtrage est très simple : l'intensité du pixel (i,j) (donc chaque élément) de l'image *filtrée* (nouvelle image) est égal à cinq fois l'intensité du pixel (i,j) de l'image de *départ* moins la somme des intensités des quatre pixels adjacents de l'image de *départ* (attention au bord de l'image – voir ci-dessous).

Ces coefficients peuvent être représentés par l'exemple de droite (où 5 est le coefficient du pixel considéré) :

Il faut également faire attention à deux choses :

- L'intensité de la nouvelle image doit toujours être comprise entre 0 et 255.
- Quand on arrive sur un bord, il ne faut pas tenir compte de la valeur des pixels qui se situeraient hors de la matrice. Vous pouvez considérer que leur intensité vaut zéro.

Exemple d'image :



254	107
255	165

Coefficients du filtre :

0	-1	0
-1	5	-1
0	-1	0