

Section Date	
Enseignant Matière	<b>M. MALDONADO</b> <b>MATHEMATIQUES APPLIQUEES</b>

## EXO : CONTRASTE D'IMAGE

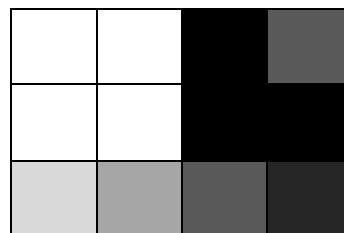
### Description

Une image en noir et blanc, découpée en pixels, est considérée comme une matrice comportant des entiers de 0 à 100, appelés dans la suite « saturation ». 0 correspond à une case blanche, 100 à une case noire, un entier entre 0 et 100 à une case grise, d'autant plus foncée qu'elle est proche de 100.

Par exemple la matrice :

$$\begin{pmatrix} 0 & 0 & 100 & 50 \\ 0 & 0 & 100 & 100 \\ 20 & 35 & 50 & 75 \end{pmatrix}$$

correspond à l'image :



## Algorithme principal

L'algorithme et la traduction en Python du programme principal sont :

### ALGO

M, Mcontraste : matrice  
nombreDeLignes, nombreDeColonnes, luminosite : entiers

#### Début

$M \leftarrow \begin{bmatrix} 0,0,0,100,75,50,0 \\ 0,0,100,100,70,40,0 \\ 0,100,100,100,65,35,0 \\ 0,100,0,100,60,30,0 \\ 0,0,0,100,55,25,0 \\ 0,0,0,100,50,20,0 \end{bmatrix}$

*// Ecriture de la matrice*

Ecrire (La matrice à traiter est :)

ecrire\_matrice (M)

*// Calcul de la luminosité*

luminosite  $\leftarrow$  calculer\_luminosite (M)

Ecrire (la luminosité de la matrice est , luminosite)

*// Calcul du contraste*

Mcontraste  $\leftarrow$  calculer\_contraste (M, luminosite)

Ecrire (La matrice contraste est :)

ecrire\_matrice (MContraste)

#### Fin

### PYTHON

*# Fonctions*

*# Programme principal*

M =  $\begin{bmatrix} 0,0,0,100,75,50,0 \\ 0,0,100,100,70,40,0 \\ 0,100,100,100,65,35,0 \\ 0,100,0,100,60,30,0 \\ 0,0,0,100,55,25,0 \\ 0,0,0,100,50,20,0 \end{bmatrix}$

*# Ecriture de la matrice*

print ("La matrice à traiter est :")

ecrire\_matrice (M)

*# Calcul de la luminosité*

luminosite = calculer\_luminosite (M)

print ("la luminosité de la matrice est ", luminosite)

*# Calcul du contraste*

Mcontraste = calculer\_contraste (M, luminosite)

print ("La matrice contraste est : ")

ecrire\_matrice (MContraste)

## PARTIE A – 30 minutes

A traiter sur feuille et à rendre avant d'accéder aux machines

### Exercice 1

Ecrire une fonction `ecrire_matrice(M)` dont le paramètre est la matrice M. Cette fonction affichera la matrice M ligne par ligne.

Exemple :  $M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$   $\begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{bmatrix}$   
`ecrire_matrice (M)` affichera :

### Exercice 2

La luminosité d'une image est la moyenne m de ses saturations.

Pour l'image de la page précédente,  $m = (100+50+100+100+20+35+50+75) / (4 \times 3) = 44$

Ecrire une fonction `calculer_luminosite (M)` dont le paramètre est la matrice M. Cette fonction renvoie la luminosité de l'image correspondant à la matrice M.

### Exercice 3

On cherche ici à accentuer les contrastes : une couleur foncée sera encore plus foncée, une couleur claire sera encore plus claire.

Pour cela, on utilise la luminosité calculée précédemment et pour chaque saturation de la matrice M :

- On la divise par 2 si cette saturation est inférieure à m
  - On la multiplie par 2 sinon, sans dépasser 100.
- Si la nouvelle saturation dépasse 100, elle prendra la valeur 100.

Avec  $m=44$ , on obtient :

$\begin{pmatrix} 0 & 0 & 100 & 50 \\ 0 & 0 & 100 & 100 \\ 20 & 35 & 50 & 75 \end{pmatrix}$  la matrice contraste vaut :  $\begin{pmatrix} 0 & 0 & 100 & 100 \\ 0 & 0 & 100 & 100 \\ 10 & 17 & 100 & 100 \end{pmatrix}$

Ecrire une fonction `calculer_contraste(M, luminosite)` dont les paramètres sont la matrice M et sa luminosité. Cette fonction renvoie la matrice contraste.

## PARTIE B – 30 minutes

A traiter sur un ordinateur en utilisant le langage Python

Vous enregistrerez votre travail sous votre nom.

### Exercice 4

Coder la fonction `ecrire_matrice(M)` de l'exercice 1 puis la tester

Aide : En Python, on itère sur les lignes d'une matrice M par `for ligne in M`  
 puis sur les colonnes d'une ligne par `for colonne in ligne`

### Exercice 5

Coder la fonction `calculer_luminosite(M)` de l'exercice 2 puis la tester

### Exercice 6

Coder la fonction `calculer_contraste(M, luminosite)` de l'exercice 3 puis la tester

### Exercice 7

Télécharger le programme principal.

Faites un copier/coller sur votre page contenant les fonctions puis les tester.

- `a / b` calcule le quotient entier de la division de a par b  
Exemple : `13 / 5` donne 2
- `a % b` calcule le reste de la division de a par b  
Exemple : `13 % 5` donne 3
- `float(a) / b` calcule le quotient de a par b  
Exemple : `float(13) / 5` donne 2,6
- `Math.Pow(a,b)` calcule a à la puissance b  
Exemple : `Math.Pow(2,3)` donne 8
- Vrai et Faux sont les booléens `true` et `false`
- Déclaration et instantiation d'une liste (d'entiers) :  
`List<int> liste = new List<int> ();`
- `liste[i]` restitue l'élément de la liste de rang i  
Exemple : `liste = [10, 25, 33, 4, 8]` `liste[1]` renvoie 25
- - Déclaration et instantiation d'une matrice (d'entiers):  
Exemple : `int [,] M = new int [3,4] ;`  
  
- Initialisation d'une matrice  
Exemple :  

<code>M = new int[3,5]{</code>	<code>{1,2,3,4,5},</code>		<code>1</code>	<code>2</code>	<code>3</code>	<code>4</code>	<code>5</code>
	<code>{6,7,8,9,10},</code>	→	<code>6</code>	<code>7</code>	<code>8</code>	<code>9</code>	<code>10</code>
	<code>{11,12,13,14,15}}</code>		<code>11</code>	<code>12</code>	<code>13</code>	<code>14</code>	<code>15</code>
- - `M[1,2]` est la valeur de la deuxième ligne, troisième colonne, soit 8
- `liste.Count` restitue le nombre d'éléments de la liste  
Exemple : `liste = [10, 25, 33, 4, 8]` `liste.Count` renvoie 5
- `liste.Add(a)` ajoute l'élément a à la liste liste  
Exemple : `liste = [10, 25, 33, 4, 8]` `liste.Add (100)` renvoie `[10, 25, 33, 4, 8, 100]`
- `liste.Insert(i, a)` ajoute l'élément a au rang i de la liste liste  
Exemple : `liste = [10, 25, 33, 4, 8]` `liste.Insert (0,100)` renvoie `[100,10, 25, 33, 4, 8]`
- La boucle Pour i allant de 1 à n se traduit par : `for (int i=1 ; i<=n ; i++)`  

...

 Pour itérer sur une liste (d'entier): `foreach (int i in liste)`  

...
- Chaîne de caractères. Comme pour les listes :  
 - `ch.Count` donne le nombre de caractères de la chaîne ch  
 - `ch[0]` est le premier caractère de la chaîne ch  
 Exemple : `ch = « AMI »`, `ch.Count` renvoie 3, `ch[0]` renvoie « A », `ch + «S»` renvoie « AMIS »
- Syntaxe pour l'écriture d'une fonction :  
Exemple :  

```
static int somme (a,b)
{
    int s = a+b ;
    return s;
}
```

- `a // b` calcule le quotient entier de la division de a par b  
Exemple : `13 // 5` donne 2
- `a % b` calcule le reste de la division de a par b  
Exemple : `13 % 5` donne 3
- `a / b` calcule le quotient de a par b  
Exemple : `13 / 5` donne 2,6
- `a ** b` calcule a à la puissance b  
Exemple : `2 ** 3` donne 8
- Vrai et Faux sont les booléens `True` et `False`
- `liste[i]` restitue l'élément de la liste de rang i  
Exemple : `liste = [10, 25, 33, 4, 8]` `liste[1]` renvoie 25
- une matrice est une liste de listes  
Exemple :  

		1	2	3	4	5	
- M =	[[1,2,3,4,5], [6,7,8,9,10], [11,12,13,14,15]]	→	6	7	8	9	10
			11	12	13	14	15

  
- `M[0]` est la première ligne de la matrice : `[1, 2, 3, 4, 5]`  
- `M[1][2]` est la valeur de la deuxième ligne, troisième colonne, soit 8
- `len(liste)` restitue le nombre d'éléments de la liste  
Exemple : `liste = [10, 25, 33, 4, 8]` `len(liste)` renvoie 5
- `liste.append(a)` ajoute l'élément a à la liste liste  
Exemple : `liste = [10, 25, 33, 4, 8]` `liste.append(100)` renvoie `[10, 25, 33, 4, 8, 100]`
- `liste.insert(i, a)` ajoute l'élément a au rang i de la liste liste  
Exemple : `liste = [10, 25, 33, 4, 8]` `liste.insert(0,100)` renvoie `[100,10, 25, 33, 4, 8]`
- `range(a,b)` est la liste entiers compris entre a et b, b non compris  
Exemple :  
- `range(0,5)` renvoie la liste `[0, 1, 2, 3, 4]`  
- `for i in range(0,5) :` correspond à : Pour i allant de 0 à 4
- Chaîne de caractères. Comme pour les listes :  
- `len(ch)` donne le nombre de caractères de la chaîne ch  
- `ch[0]` est le premier caractère de la chaîne ch  
Exemple : `ch = « AMI »`, `len(ch)` renvoie 3, `ch[0]` renvoie « A », `ch + «S»` renvoie « AMIS »
- Syntaxe pour l'écriture d'une fonction :  
Exemple :  

<pre>&gt;&gt;&gt; def somme_produit(a,b) :     somme = a+b     produit = a*b     return somme, produit</pre>	donne	<pre>&gt;&gt;&gt; somme_produit (3,4) (7,12) &gt;&gt;&gt; a,b = somme_produit (3,4) &gt;&gt;&gt; a 7 &gt;&gt;&gt; b 12</pre>
--	-------	--