

**BREVET DE TECHNICIEN SUPÉRIEUR**  
**SERVICES INFORMATIQUES AUX ORGANISATIONS**

**SESSION 2012**

**Sous-épreuve E 22 Algorithmique appliquée**

- Il est formellement interdit d'utiliser toute connexion à un réseau interne ou externe au centre d'examen, quel qu'en soit le procédé.
- L'usage d'une calculatrice est autorisée.
- Cette épreuve comporte 2 parties :
  - une première partie qui dure 30 minutes à l'issue de laquelle vous devez fournir une production écrite répondant au sujet.
  - une deuxième partie qui dure également 30 minutes. Vous travaillerez sur un des ordinateurs d'examen pour coder les algorithmes papiers sous Python.  
Vous enregistrerez votre travail sur une clé USB fournie par votre professeur.  
Vous imprimerez votre travail et le signerez.
- Vous n'oubliez pas de rendre :
  - Le sujet.
  - L'algorithme papier où figureront votre nom et prénom.
  - L'impression de votre programme signée

Nom :

Prénom :

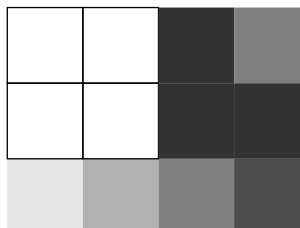
**Problème :**

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 que le nombre 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 :



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

**Variables :**

- M, Mcontraste, : liste de listes
- nombredelignes, nombredecolonnes, luminosité : entiers

**Début**

```
M ← [[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]]
```

```
# Écriture de la matrice
```

```
Écrire La matrice à traiter est :
```

```
ecrire_matrice(M)
```

```
# Calcul de la luminosité
```

```
luminosite ← luminosite(M)
```

```
Écrire La luminosité de la matrice est
```

```
print(luminosite)
```

```
# Calcul de la matrice contraste
```

```
Mcontraste ← contraste(M,luminosite)
```

```
Écrire La matrice contraste est :
```

```
ecrire_matrice(Mcontraste)
```

**Fin****# Fonctions**

```
#Programme principal
```

```
M=[[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]]
```

```
print("La matrice à traiter M est : ")
```

```
ecrire_matrice(M)
```

```
luminosite=luminosite(M)
```

```
print("La luminosité de la matrice est ")
```

```
print(luminosite)
```

```
Mcontraste=contraste(M,luminosite)
```

```
print("La matrice contraste est :")
```

```
ecrire_matrice(Mcontraste)
```

## Partie A :

8 points

A traiter sur feuille et à rendre avant d'accéder aux machines.  
Durée maximum : 30 minutes

**Exercice 1**

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

*Exemple :*

```
>>> MAT=[[1,2,3],[4,5,6],[7,8,9]]
>>> écrire_matrice(MAT)
```

```
[[1, 2, 3]
 [4, 5, 6]
 [7, 8, 9]]
```

**Exercice 2**

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

Pour l'image ci-dessus  $m = \frac{100 + 50 + 100 + 100 + 20 + 35 + 50 + 75}{4 \times 3} = 44$

Écrire une fonction `luminosite(MAT)`, dont le paramètre est la matrice `MAT`. Cette fonction renvoie la luminosité de l'image correspondant à la matrice `MAT`.

**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  $m$  calculée précédemment et pour chaque saturation de la matrice `MAT`

- 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 :

$$\begin{pmatrix} 0 & 0 & 100 & 50 \\ 0 & 0 & 100 & 100 \\ 20 & 35 & 50 & 75 \end{pmatrix} \text{ la matrice de contraste } \implies \begin{pmatrix} 0 & 0 & 100 & 100 \\ 0 & 0 & 100 & 100 \\ 10 & 17 & 100 & 100 \end{pmatrix}$$

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

**Partie B :****8 points**

A traiter sur un ordinateur en utilisant le langage Python.  
Vous enregistrerez votre travail sous votre nom.  
Durée maximum : 30 minutes

**Exercice 4**

Coder sous Python la fonction `ecrire_matrice(MAT)` de l'exercice 1 puis la tester.

**Exercice 5**

Coder sous Python la fonction `luminosite(MAT)` de l'exercice 2 puis la tester.

**Exercice 6**

Coder sous Python la fonction `contraste(MAT,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 le tester.

**BREVET DE TECHNICIEN SUPÉRIEUR**  
**SERVICES INFORMATIQUES AUX ORGANISATIONS**

**Aide mémoire**

- `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 le produit de `a` par `b`.  
Exemple : `3 * 5` donne 15
- `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 :
 

– <code>M=[[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15]]</code>	<code>==&gt;</code>	<table style="border-collapse: collapse;"> <tr><td style="padding-right: 10px;">1</td><td style="padding-right: 10px;">2</td><td style="padding-right: 10px;">3</td><td style="padding-right: 10px;">4</td><td style="padding-right: 10px;">5</td></tr> <tr><td style="padding-right: 10px;">6</td><td style="padding-right: 10px;">7</td><td style="padding-right: 10px;">8</td><td style="padding-right: 10px;">9</td><td style="padding-right: 10px;">10</td></tr> <tr><td style="padding-right: 10px;">11</td><td style="padding-right: 10px;">12</td><td style="padding-right: 10px;">13</td><td style="padding-right: 10px;">14</td><td style="padding-right: 10px;">15</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4	5													
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 la longueur 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 des 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 la longueur de la chaîne de caractères `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>
--	-------	---