
BTS SIO

SOUS-ÉPREUVE E22

ALGORITHMIQUE APPLIQUÉE

CONTRÔLE EN COURS DE FORMATION

Déroulement de l'épreuve

Cette épreuve de Contrôle en cours de Formation (CCF) se déroule en trois étapes :

– **Étape 1 : Écrit (30 minutes)**

Vous devez traiter la partie A du sujet. Pour cette partie, l'ordinateur est interdit mais la calculatrice est autorisée.

Vous inscrirez vos réponses dans le document réponse à la fin du sujet.

Les algorithmes à écrire peuvent être rédigés en **langage naturel** ou en PYTHON mais ni en C# ni en VB.NET.

À la fin de l'étape 1, votre document réponse doit être remis à la personne surveillant l'épreuve. Vous garderez le sujet.

– **Étape 2 : sur machine (30 minutes)**

Vous devez traiter la partie B du sujet à l'aide d'un ordinateur. Le langage utilisé est celui travaillé dans l'année, à savoir PYTHON. Vous sauvegarderez votre travail sur la clé USB fournie.

La durée totale pour effectuer les deux premières étapes est exactement d'une heure.

– **Étape 3 : oral (20 minutes au maximum)**

Cette partie se déroule en deux temps. Tout d'abord, vous disposez de 10 minutes pour présenter votre travail de l'étape 2 puis, au cours des 10 minutes suivantes, un entretien permet de préciser votre démarche.

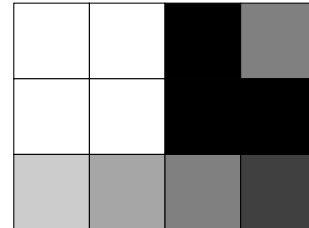
À la fin de l'épreuve le sujet devra être rendu à l'examineur.

Une image en niveaux de gris est composée de pixels et peut être représentée *mathématiquement* par une matrice **saturation** dont chaque coefficient est un entier entre 0 et 255.

Plus le coefficient est petit, plus le gris est sombre, ainsi 0 correspond à un pixel noir et 255 à un pixel blanc.

La matrice

$$\begin{pmatrix} 255 & 255 & 0 & 128 \\ 255 & 255 & 0 & 0 \\ 204 & 165 & 128 & 64 \end{pmatrix}$$



correspond à l'image ci-contre.

Étape 1

Voici l'algorithme d'une fonction qui crée une matrice 3×4 (3 lignes de 4 colonnes).

La matrice est représentée par une liste de listes, chacun de ses coefficients est saisi par l'utilisateur. La fonction

- ne prend rien en entrée;
- renvoie une liste de listes d'entiers.

Algorithme

```
fonction saturation_matrice()
    variables
        i,j, nombre : entiers # i compte les lignes, j les colonnes
        saturation : liste # en fait liste de listes d'entiers

    saturation ← [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
    pour i allant de 0 à ..... faire
        .....
        tant que nombre > 255 ou nombre < 0 faire
            lire nombre
        fin tant que
        saturation[i][j] ← nombre
    fin pour
fin pour
renvoyer saturation
```

Question 1

Compléter les pointillés pour que la fonction remplisse son rôle.

La fonction suivante, nommée `luminosite` :

- prend en entrée une liste de listes d'entiers `matrice`, qui représente une matrice 3×4 ;
- renvoie un entier `lumi`.

Algorithme

```
fonction luminosite(matrice)

    variables
        i, j, somme, luminosite : entiers

    somme ← 0
    pour i allant de 0 à 2 faire
        pour j allant de 0 à 3 faire
            somme ← somme + matrice[i][j]
        fin pour
    fin pour
    lumi ← partie_entiere(somme/12) # exemple : partie_entiere(3.4)=3
    renvoyer lumi
```

Question 2

Lorsque `M = [[0, 0, 100, 50], [0, 70, 100, 100], [20, 35, 50, 75]]`,
quelle est la valeur de `luminosite(M)` ?

Pour accentuer le contraste de l'image représentée par une matrice `M`, on modifie chacune des valeurs de cette matrice comme ceci :

- d'abord on utilise la valeur `luminosite(M)` calculée précédemment que l'on note `lumi`.
- Pour chacune des 12 valeurs de la matrice `M`
 - si elle est inférieure ou égale à `lumi` on la divise par 2 et on garde la partie entière du résultat;
 - sinon on la multiplie par 2 sans dépasser 255 (si la nouvelle valeur dépasse 255, on la ramène à 255).

Question 3

Écrire l'algorithme (ou le code PYTHON) de la fonction `contraste` qui

- prend en entrée une liste de listes d'entiers `matrice`, qui représente une matrice 3×4 ;

- renvoie une *autre* matrice : la matrice d'entrée avec un contraste accentué.

Étape 2

Le fichier `luminosite.py` contient une variable `M` représentant la matrice de la question 2 de l'étape 1.

Question 4

Implémenter la fonction `luminosite`.
Pour prendre la partie entière d'une valeur `v`, on pourra utiliser `int(v)`.

Question 5

Implémenter la fonction `contraste`.

Question BONUS

Implémenter la fonction `matrice_aleatoire` qui

- en entrée prend 2 entiers `n` et `p`;
- renvoie une matrice $n \times p$ dont tous les coefficients sont des nombres aléatoires entre 0 et 255.

On utilisera `from random import randint`.
`randint(a,b)` renvoie un nombre entier au hasard, compris entre `a` et `b`.