

---

# 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 :

– **Écrit (30 minutes)**

Vous devez traiter l'étape 1 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.

– **Machine (30 minutes)**

Vous devez traiter l'étape 2 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.

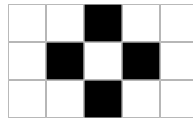
– **Oral (20 minutes au maximum)**

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 rectangulaire en noir et blanc peut être représentée par une liste de lignes qui sont des listes d'entiers valant 0 (pour le noir) et 1 (pour le blanc).

Par exemple l'image suivante, de dimensions  $5 \times 4$



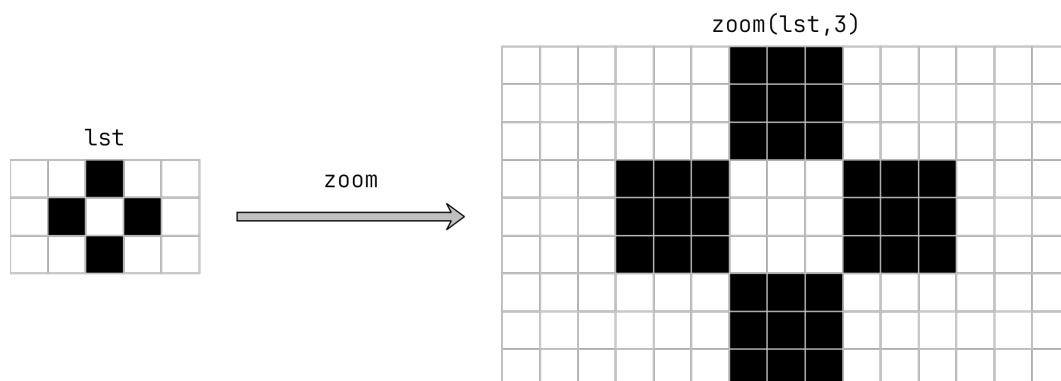
est représentée par la liste suivante :

```
[[0, 0, 1, 0, 0],
 [0, 1, 0, 1, 0],
 [0, 0, 1, 0, 0],
 [0, 1, 0, 1, 0],
 [0, 0, 1, 0, 0]]
```

On aimerait construire une fonction **zoom** qui

- en entrée prend une liste `lst` qui représente une image rectangulaire et un `int` strictement positif `k`;
- renvoie une liste qui correspond à l'image représentée par `lst` grossie d'un facteur `k`.

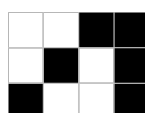
Ci-dessous figure un exemple d'utilisation de la fonction **zoom**



## Étape 1

### Question 1

Dessiner l'image obtenue en appliquant `zoom(lst, 2)` avec une liste `lst` représentant l'image suivante :



## Question 2

Si `lst` représente une image de  $n$  lignes par  $p$  colonnes, et que `lst2 = zoom(lst, k)`, quelle est la taille de

1. `lst2`?
2. `lst2[0]`?

## Question 3

Compléter le pseudocode de la fonction `zoom_horiz` qui

- en entrée prend une liste d'entiers `ligne` et un entier `k`;
- renvoie une liste d'entiers dans laquelle chaque valeur de `ligne` est dupliquée `k` fois.

```
fonction zoom(ligne, k)

    variables
        résultat : liste
        valeur, compteur, i : entiers

    résultat ← liste vide
    n ← longueur(ligne)
    pour i ...
        pour j ...
            ajouter ... à la fin de résultat
    renvoyer résultat
```

## Question 4

Compléter le pseudocode de la fonction `zoom` que l'on veut coder

```
fonction zoom(lst, k)

    variables
        résultat : liste
        valeur, compteur, i : entiers

    résultat ← liste vide
    p ← longueur(lst)
    pour i ...
        pour j ...
            ajouter ... à la fin de résultat
```

renvoyer résultat

## Étape 2

### Question 5

Ouvrir le fichier `zoom.py` et coder les fonctions manquantes.

### Question 6

Coder la fonction `affiche` qui

- en entrée prend une liste `lst` qui représente une image;
- ne renvoie rien mais affiche joliment l'image avec des `'.'` à la place des zéros et des `'*'` à la place des 1.

On rappelle que `print('*', end="")` affiche `'*'` sans retour à la ligne.