

---

# **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)**

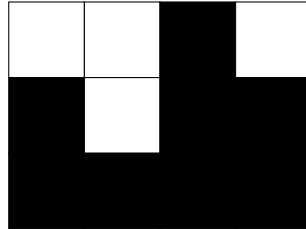
Cette partie se déroule en deux temps. Tout d'abord, vous disposez de 10 minutes pour présenter votre travail de l'étape 2 puis, au cours des 10 minutes suivantes, un entretien permet de préciser votre démarche.

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

# Compression RLE d'une image en noir et blanc

Une image en noir et blanc peut être représentée par une liste d'entiers valant 0 (pour le noir) et 1 (pour le blanc).

Par exemple l'image suivante, de dimensions  $3 \times 4$



est représentée par la liste obtenue en parcourant les pixels de gauche à droite et du haut vers le bas :

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

C'est une liste de longueur 12 car l'image comporte  $3 \times 4 = 12$  pixels.

L'objectif de ce sujet est de compresser l'image, c'est-à-dire d'obtenir une liste plus courte qui nous permet de retrouver l'image. Pour simplifier l'image sera toujours de dimension  $3 \times 4$ .

On utilise la méthode suivante :

- on commence au début de la liste et on compte les zéros;
- puis on compte le nombre de uns;
- et ainsi de suite jusqu'à la fin de la liste.

Par exemple avec la liste précédente :

- on commence par compter les zéros : il y en a... zéro!
- puis 2 uns;
- puis 1 zéro;
- puis 1 un;
- et ainsi de suite.

On obtient la liste compressée suivante :

[0, 2, 1, 1, 1, 1, 6]

Elle est bien plus courte que la liste de départ car sa longueur est 7.

### Définition

Le taux de compression  $T$  (en pourcentage) d'une liste est défini par

$$T = 100 \times \left( 1 - \frac{\text{longueur de la liste compressée}}{\text{longueur de la liste}} \right)$$

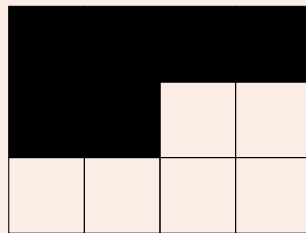
Par exemple le taux de compression dans la situation précédente est

$$100 \times \left( 1 - \frac{7}{12} \right) \approx 41,6\%$$

## Étape 1

### Question 1

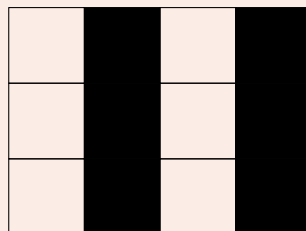
Pour l'image suivante :



- Donner la liste associée.
- Donner la liste compressée.
- Quel est le taux de compression ?

### Question 2

Refaire la même chose avec l'image suivante :



L'algorithme de compression est en partie donné :

## Langage naturel

```
1 fonction compresse(ligne)
2
3   variables
4     résultat : liste
5     valeur, compteur, i : entiers
6
7   résultat ← liste vide
8   valeur ← 0           # on part de la valeur 0
9   compteur ← 0         # au départ il y en a zéro
10  i ← 0                 # on commence au début de la ligne
11  tant que i < longueur(ligne)
12    si ligne[i] = .....
13      compteur ← compteur + 1
14    sinon
15      valeur ← 1 - valeur
16      ajouter compteur à la fin de résultat
17      compteur ← 1
18      .....
19  ajouter compteur à la fin de résultat
20  renvoyer résultat
```

### Question 3

- Si, juste avant d'exécuter la ligne 15, **valeur** vaut zéro, que devient **valeur** après l'exécution de la ligne 15?
- Même question si **valeur** vaut 1.
- Quel est le rôle de cette ligne?

### Question 4

Compléter l'algorithme.

### Question 5

Dessiner l'image qui produit la liste compressée

[3,3,3,3]

## Étape 2

Le fichier `r1e.py` l'implémentation en PYTHON de la fonction **compresse** ainsi que 2 listes de test.

### Question 7

Implémenter la fonction **taux\_de\_compression** qui

- en entrée prend une liste composée de 0 et de 1;
- renvoie le taux de compression quand on compresse la liste.

### Question 8

Implémenter la fonction **decompresse** qui

- en entrée prend une liste compressée;
- renvoie la liste décompressée.