

## Partie 1 : Tableaux 1D - Accès par indice

### 1.1 Exercice 1.1 : Gestion d'un inventaire de jeu vidéo

Vous développez un jeu de rôle où le personnage possède un inventaire de 10 emplacements. Chaque emplacement contient un objet représenté par une chaîne de caractères.

```
1 inventaire = ["épée", "potion", "bouclier", "vide", "clé", "vide",  
  , "arc", "carte", "vide", "torche"]
```

#### Exemple 1 : Accéder à un élément

```
1 # Pour afficher l'objet à l'indice 2  
2 print(inventaire[2]) # Affiche : bouclier
```

#### Exemple 2 : Modifier un élément

```
1 # Pour remplacer l'objet à l'indice 3  
2 inventaire[3] = "diamant"  
3 print(inventaire[3]) # Affiche : diamant
```

#### Exemple 3 : Échanger deux éléments

```
1 # Pour échanger les éléments aux indices 1 et 5  
2 temp = inventaire[1]  
3 inventaire[1] = inventaire[5]  
4 inventaire[5] = temp
```

#### Questions :

1. Affichez l'objet situé à l'emplacement 4 de l'inventaire.
2. Le joueur ramasse un "parchemin". Remplacez le premier emplacement "vide" par "parchemin".
3. Le joueur utilise sa potion (indice 1). Remplacez-la par "vide".
4. Échangez les objets aux emplacements 0 et 6 (épée et arc).
5. Créer une fonction `echange` qui prend une liste `L` ainsi que 2 indices `i` et `j` pour paramètres, et qui modifie la liste `L` en permutant les objets aux rangs `i` et `j`. Cette fonction n'aura pas de valeur de retour.
6. Ecrire l'instruction qui échange les objets aux emplacements 0 et 6, mais cette fois-ci, en utilisant la fonction `echange`.

## 1.2 Exercice 1.2 : Thermomètre intelligent

Un thermomètre enregistre les températures de chaque heure de la journée (24 valeurs).

```
1 temperatures = [12, 11, 10, 10, 9, 9, 11, 14, 17, 20, 23, 25, 27,
2                 28, 27, 25, 22, 19, 17, 15, 14, 13, 13, 12]
```

**Exemple : Calculer un écart**

```
1 # Pour calculer l'écart entre deux températures
2 ecart = temperatures[10] - temperatures[3]
3 print("L'écart est de", ecart, "degrés")
```

**Exemple : Augmenter une valeur**

```
1 # Pour augmenter de 3 degrés la température à l'indice 5
2 temperatures[5] = temperatures[5] + 3
3 # ou plus court :
4 temperatures[5] += 3
```

**Questions :**

1. Affichez la température à 14h (indice 14).
2. Une erreur de capteur s'est produite à 8h. Modifiez la valeur à l'indice 8 pour la remplacer par 16.
3. Pourrait-on utiliser la fonction `echange` ? Si oui, écrire l'instruction correspondante.
4. Calculez l'écart entre la température maximale (indice 13) et minimale (indice 5).
5. Augmentez de 2 degrés la température de minuit (indice 0).

Partie 2

## Partie 2 : Tableaux 1D - Parcours par élément et par indice

### 2.1 Exercice 2.1 : Analyse d'une partie de Scrabble

Un joueur de Scrabble possède les lettres suivantes dans son chevalet :

```
1 chevalet = ['A', 'E', 'R', 'T', 'I', 'N', 'S']
```

**Exemple 1 : Parcours par élément**

```
1 # Afficher chaque lettre
2 for lettre in chevalet:
3     print(lettre)
```

**Exemple 2 : Compter des éléments**

```
1 # Compter les 'E' dans le chevalet
2 compteur = 0
3 for lettre in chevalet:
4     if lettre == 'E':
5         compteur += 1
6 print("Il y a", compteur, "lettre E")
```

### Exemple 3 : Vérifier une condition

```
1 # Vérifier si le chevalet contient un 'A'
2 if 'A' in chevalet:
3     print("Il y a un A !")
4 else:
5     print("Pas de A")
```

### Questions avec parcours par élément :

1. Affichez toutes les lettres du chevalet, une par ligne, en utilisant une boucle `for lettre in chevalet`.
2. Comptez combien de voyelles (A, E, I, O, U, Y) se trouvent dans le chevalet.
3. Affichez "BINGO!" si le chevalet contient au moins une lettre 'S', sinon affichez "Pas de S".
4. Ecrire une fonction `compte` qui prend une liste `L` ainsi qu'une lettre `V` en paramètre, et qui retourne le nombre de fois où cette lettre est présente dans la liste.
5. Ecrire l'instruction qui appelle cette fonction pour déterminer le nombre de E dans le chevalet  
['A', 'E', 'R', 'T', 'I', 'N', 'S']

## 2.2 Exercice 2.2 : Parcours par indice

### Exemple 1 : afficher l'indice et l'élément

```
1 # Afficher chaque lettre avec sa position
2 for i in range(len(chevalet)):
3     print("Position", i, ":", chevalet[i])
```

### Exemple 2 : Modifier en parcourant

```
1 # Remplacer tous les 'A' par des '*'
2 for i in range(len(chevalet)):
3     if chevalet[i] == 'A':
4         chevalet[i] = '*'
```

### Exemple 3 : Créer un nouveau tableau

```
1 # Créer un tableau avec uniquement les voyelles
2 voyelles_trouvees = []
3 for i in range(len(chevalet)):
4     if chevalet[i] in ['A', 'E', 'I', 'O', 'U', 'Y']:
5         voyelles_trouvees.append(chevalet[i])
6 print(voyelles_trouvees)
```

**Questions avec parcours par indice :**

4. Affichez chaque lettre avec sa position : "Position 0 : A", "Position 1 : E", etc.
5. Remplacez toutes les voyelles par des '\*' en parcourant le tableau par indice.
6. Créez un nouveau tableau contenant uniquement les consonnes en utilisant les indices.

## Partie 3

**Partie 3 : Tableaux 2D - Parcours par élément et par indice****3.1 Exercice 3.1 : Grille de Morpion****3.1.1 Parcours par élément**

Une grille de morpion est représentée par un tableau 2D de 3×3 :

```

1 morpion = [
2     ['X', 'O', 'X'],
3     ['O', 'X', ' '],
4     [' ', ' ', 'O']
5 ]

```

**Exemple 1 : afficher les éléments**

```

1 for ligne in morpion:
2     for case in ligne:
3         print(case, end=' ')
4     print() # retour à la ligne
5 # Affiche toutes les valeurs ligne par ligne:
6 X | O | X |
7 O | X |   |
8   |   | O |

```

**Exemple 2 : Compter les éléments**

```

1 # Compter les 'X'
2 compteur = 0
3 for ligne in morpion:
4     for case in ligne:
5         if case == 'X':
6             compteur += 1
7 print("Nombre de X :", compteur)

```

**Questions :**

1. Affichez tous les 'X' de la grille, ligne par ligne en utilisant deux boucles `for`.
2. Comptez le nombre de cases vides (contenant ' ').
3. Déterminez qui a le plus de coups joués (X ou O).

### 3.1.2 parcours par indice :

#### Exemple 1 : Parcours par indice (tableau 2D)

```

1 # Afficher la grille avec les indices
2 for i in range(len(morpion)):
3     for j in range(len(morpion[i])):
4         print(morpion[i][j], end=' ')
5     print()

```

#### Exemple 2 : Accéder à un élément précis

```

1 # Afficher et modifier un élément
2 print("Case [0][1] :", morpion[0][1]) # Affiche 'O'
3 morpion[0][1] = 'X' # Modifier la case

```

#### Exemple 3 : Vérifier une ligne

```

1 # Vérifier si la ligne 0 contient trois symboles identiques
2 ligne = morpion[0]
3 if ligne[0] == ligne[1] == ligne[2] and ligne[0] != ' ':
4     print("La ligne 0 est gagnante !")

```

#### Questions :

1. Le joueur O joue en position [1][2] (ligne 1, colonne 2). Ecrire l'instruction qui modifie la grille.
2. Quelles sont les coordonnées `morpion[i][j]` pour chacune des cases? Dessinez sur votre cahier une matrice pour placer les coordonnées `[i][j]` dans chaque case..
3. Ecrire les instructions qui vérifient si la première ligne contient trois symboles identiques (condition de victoire).
4. Ecrire une fonction `ligne_X` qui prend pour paramètre une matrice `M` comme la grille de morpion (3\*3). `M` contient dans chaque case 'X', 'O', ou ' '. Cette fonction retournera `True` si l'une des lignes est remplie avec trois 'X'.

### 3.2 Exercice 3.2 : Grille de Sudoku (9×9)

Voici une grille de Sudoku partiellement remplie :

```

1 sudoku = [
2     [5, 3, 0, 0, 7, 0, 0, 0, 0],
3     [6, 0, 0, 1, 9, 5, 0, 0, 0],
4     [0, 9, 8, 0, 0, 0, 0, 6, 0],
5     [8, 0, 0, 0, 6, 0, 0, 0, 3],
6     [4, 0, 0, 8, 0, 3, 0, 0, 1],
7     [7, 0, 0, 0, 2, 0, 0, 0, 6],
8     [0, 6, 0, 0, 0, 0, 2, 8, 0],
9     [0, 0, 0, 4, 1, 9, 0, 0, 5],
10    [0, 0, 0, 0, 8, 0, 0, 7, 9]
11 ]

```

Les cases vides sont représentées par 0.

**Exemple : Compter les cases vides**

```

1 # Compter les 0 dans toute la grille
2 compteur = 0
3 for i in range(len(sudoku)):
4     for j in range(len(sudoku[i])):
5         if sudoku[i][j] == 0:
6             compteur += 1
7 print("Nombre de cases vides :", compteur)

```

**Exemple : Afficher une ligne**

```

1 # Afficher la ligne 0
2 for j in range(len(sudoku[0])):
3     print(sudoku[0][j], end=' ')
4 print()

```

**Exemple : Afficher une colonne**

```

1 # Afficher la colonne 2
2 for i in range(len(sudoku)):
3     print(sudoku[i][2], end=' ')
4 print()

```

**Exemple : Modifier une case**

```

1 # Remplacer le 0 en position [0][2] par 4
2 sudoku[0][2] = 4
3 print("Nouvelle valeur :", sudoku[0][2])

```

**Exemple : Extraire un bloc 3×3**

```

1 # Extraire le bloc en haut à gauche (lignes 0-2, colonnes 0-2)
2 bloc = []

```

```
3 for i in range(0, 3): # lignes 0, 1, 2
4     ligne_bloc = []
5     for j in range(0, 3): # colonnes 0, 1, 2
6         ligne_bloc.append(sudoku[i][j])
7     bloc.append(ligne_bloc)
8 print(bloc)
```

### Exemple : Vérifier les doublons dans une ligne

```
1 # Vérifier si la ligne 1 contient des doublons (sans compter les
   0)
2 ligne = sudoku[1]
3 nombres = []
4 for valeur in ligne:
5     if valeur != 0:
6         if valeur in nombres:
7             print("Doublon détecté !")
8         else:
9             nombres.append(valeur)
```

### Questions avec parcours par indice :

1. Comptez le nombre de cases vides (valeur 0) dans toute la grille.
2. Affichez tous les chiffres de la première ligne (ligne d'indice 0).
3. Affichez tous les chiffres de la cinquième colonne (colonne d'indice 4).
4. Remplacez le 0 en position [0][2] par le chiffre 4.
5. Ecrire une fonction *doublons*, qui prend en paramètre une grille de sudoku G, ainsi qu'un numero de ligne i, et qui retourne un booléen True/False. selon si la ligne i contient des *doublons*. (sans compter les 0).
6. Appeler la fonction *doublons* pour vérifier si la première ligne contient des *doublons*.
7. Créez une fonction *bloc* qui extrait l'un des 9 blocs 3×3, n'importe où dans la grille. (par exemple, le bloc en haut à gauche : lignes 0-2, colonnes 0-2), à partir des indices i, j de départ (coin supérieur gauche du bloc).