

Laboratoire 2

Partie I : Matrices (10 points)

Implémentez les fonctions suivantes :

Q1. AfficheContenu(UneMatrice) qui permet d'afficher le contenu d'une matrice **Unematrice**.

Q2. Recherche(element,Matrice) qui permet de rechercher un élément dans une matrice. Cette fonction doit retourner un booléen.

Q3. Occurrence(element,Matrice) qui permet de compter le nombre d'occurrence d'un élément dans une matrice.

Q4. Indices(element,Matrice) qui permet de retourner une liste des indices (x,y) des occurrences d'un élément. x et y donnent respectivement les informations sur la ligne et la colonne.

Q5. MinMatrice(UneMatrice) qui permet de retourner la valeur minimale d'une matrice.

Q6. MaxMatrice(UneMatrice) qui permet de retourner la valeur maximale d'une matrice.

Q7. MoyenneMatrice(UneMatrice) qui permet de retourner la moyenne d'une matrice.

Dans cette partie, vous devez appeler les fonctions définies dans la **partie I** selon les scénarios suivants :

Q8. Initialiser les valeurs de la matrice **Mat** comme suit : **[[2 3 2 3],[7 8 8 9],[9 6 3 3]]**.

Q9. Afficher le contenu de la matrice **Mat**.

Q10. Compter et afficher le nombre d'occurrence de 3 dans **Mat**.

Q11. Compter et afficher le nombre d'occurrence de -99 dans **Mat**.

Q12. Déterminer et afficher la valeur minimale de **Mat**.

Q13. Déterminer et afficher la valeur maximale de **Mat**.

Q14. Calculer et afficher la moyenne de **Mat**.

Partie II : map, filter, sorted et lambda (10 points)

Q15. À partir de la liste de nombres suivante :

```
uneListe = [233.344, 21.903, 143.352, 78.109, 35.566]
```

Faites un programme qui utilise la fonction ***filter()*** pour conserver uniquement les valeurs de cette liste situées entre 0 et 100.

Q16. À partir de la liste de mots suivante :

```
uneListe = ['Une', 'fois', 'l', 'installation', 'terminée,', 'vous', 'pouvez', 'vous', 'connecter',  
'à', 'OpenSSH', 'Server', 'à', 'partir', 'd', 'un', 'appareil', 'Windows', '10', 'ou', 'Windows',  
'Server', '2019']
```

Faites un programme qui utilise la fonction ***filter()*** avec une fonction *lambda* pour éliminer les mots de deux lettres ou moins.

Q17. À partir de la liste de nombres suivante :

```
uneListe = [2.3, 12.8, 6.7, 31.2, 3.0]
```

Ces nombres représentent des valeurs de rayons d'un cercle. Faites un programme qui utilise la fonction ***map()*** avec une fonction *lambda* pour retourner une liste des circonférences calculées à partir de ces valeurs (la circonférence = $2 * \text{PI} * \text{rayon}$).

Q18. À partir de la liste de mots suivante :

```
uneListe = ['vous', 'pouvez', 'vous', 'connecter']
```

Faites un programme qui utilise la fonction ***map()*** pour retourner une liste des mots inversés ('sou', 'zevuop', etc.)

Q19. À partir d'une liste d'adresses MAC séparées par des retours de chariot ("`\n`"), de différents formats :

```
s = "d2-7c-2F-8A-3E-19\nbf:a0:de:8d:82:11\nE2:10:71:A4:58:28\n86-56-f2-5f-82-da\n52-4a-c1-42-72-dd\n8A:02:38:15:14:67"
```

Faite un programme qui :

1. Élimine les adresses débutant par "08:00"
2. Formate les adresses en utilisant des minuscules et le caractères ":" pour séparer les octets (e.g. 12:34:56:78:90:ab)
3. Affiche les résultats dans une liste.

Vous devez utiliser les fonctions *lambda*, *map()*, *split()* et *join()* lorsque c'est approprié.