

QCM de Programmation - Universitaire et Prépa Concours

1. Concepts avancés en programmation

1. Quel est le résultat de l'exécution du code suivant en Python ?

```
```python
def func(a, b=[]):
 b.append(a)
 return b
```

```
result1 = func(1)
```

```
result2 = func(2)
```

```
print(result1)
```

```
print(result2)
```

```
...
```

A. [1] [2]

B. [1] [1, 2]

C. [1, 2] [1, 2]

D. [1] [1, 2]

*Réponse: C. [1, 2] [1, 2]*

*Explication: La liste par défaut `b` est évaluée une seule fois lors de la définition de la fonction, et non à chaque appel. Ainsi, les éléments sont ajoutés à la même liste lors des appels successifs.*

2. Lequel des éléments suivants est vrai concernant la gestion de la mémoire en Python ?

A. Python utilise le ramasse-miettes pour la gestion de la mémoire

B. Les objets non référencés sont immédiatement supprimés

## QCM de Programmation - Universitaire et Prépa Concours

- C. Python n'a pas besoin de gestion de mémoire
- D. La mémoire doit être libérée manuellement par le programmeur

*Réponse: A. Python utilise le ramasse-miettes pour la gestion de la mémoire*

*Explication: Python utilise un ramasse-miettes (garbage collector) pour gérer automatiquement la mémoire et récupérer l'espace utilisé par les objets non référencés.*

### 2. Structures de données avancées

3. Quelle structure de données est la plus appropriée pour implémenter une file d'attente avec une complexité de  $O(1)$  pour les opérations d'enfilement et de défilement ?

- A. Liste chaînée simple
- B. Liste chaînée double
- C. Tableau dynamique
- D. Pile

*Réponse: B. Liste chaînée double*

*Explication: Une liste chaînée double permet une insertion et une suppression efficaces aux deux extrémités, ce qui est idéal pour implémenter une file d'attente avec des opérations en  $O(1)$ .*

4. Quel type de données est utilisé pour implémenter une table de hachage ?

- A. Liste
- B. Dictionnaire
- C. Ensemble
- D. Arbre binaire

*Réponse: B. Dictionnaire*

## QCM de Programmation - Universitaire et Prépa Concours

*Explication: En Python, un dictionnaire est utilisé pour implémenter une table de hachage, offrant un accès en temps constant moyen aux éléments.*

### 3. Algorithmes avancés et complexité

5. Quel algorithme de tri est le plus adapté pour trier de très grandes quantités de données déjà partiellement triées ?

- A. Tri rapide
- B. Tri fusion
- C. Tri par insertion
- D. Tri par sélection

*Réponse: C. Tri par insertion*

*Explication: Le tri par insertion est très efficace pour les données déjà partiellement triées, car il peut insérer chaque élément dans sa position correcte avec peu de comparaisons et d'échanges.*

6. Quelle est la complexité temporelle de l'algorithme de Dijkstra pour trouver le plus court chemin dans un graphe avec des arêtes pondérées positivement en utilisant un tas de Fibonacci ?

- A.  $O(V^2)$
- B.  $O(E + V \log V)$
- C.  $O(E \log V)$
- D.  $O(E^2)$

*Réponse: B.  $O(E + V \log V)$*

*Explication: En utilisant un tas de Fibonacci, l'algorithme de Dijkstra a une complexité temporelle de  $O(E + V \log V)$ , ce qui est plus efficace que l'utilisation d'une simple file de priorité.*

## QCM de Programmation - Universitaire et Prépa Concours

### 4. Programmation orientée objet avancée

7. Quelle est la différence entre une classe abstraite et une interface en programmation orientée objet ?

- A. Une classe abstraite ne peut pas avoir de méthodes concrètes
- B. Une interface peut avoir des constructeurs
- C. Une classe abstraite peut avoir des méthodes avec implémentation
- D. Une interface ne peut pas être implémentée par plusieurs classes

*Réponse: C. Une classe abstraite peut avoir des méthodes avec implémentation*

*Explication: Une classe abstraite peut contenir des méthodes avec implémentation, tandis qu'une interface ne peut contenir que des déclarations de méthodes sans implémentation.*

8. En Python, quelle méthode est appelée lorsqu'un attribut de l'objet est accédé ?

- A. `__init__`
- B. `__new__`
- C. `__getattr__`
- D. `__setattr__`

*Réponse: C. `__getattr__`*

*Explication: La méthode `__getattr__` est appelée lorsqu'un attribut d'un objet est accédé et que cet attribut n'est pas trouvé dans les méthodes habituelles d'accès aux attributs.*

### 5. Programmation fonctionnelle avancée

9. En Haskell, quelle est la fonction utilisée pour appliquer une fonction à chaque élément d'une liste ?

## QCM de Programmation - Universitaire et Prépa Concours

- A. reduce
- B. filter
- C. foldr
- D. map

*Réponse: D. map*

*Explication: La fonction `map` en Haskell est utilisée pour appliquer une fonction à chaque élément d'une liste, retournant une nouvelle liste avec les résultats.*

10. Quelle est la principale caractéristique de l'évaluation paresseuse (lazy evaluation) en programmation fonctionnelle ?

- A. Calcul immédiat des expressions
- B. Calcul différé des expressions jusqu'à ce qu'elles soient nécessaires
- C. Exécution parallèle des expressions
- D. Évaluation des expressions en arrière-plan

*Réponse: B. Calcul différé des expressions jusqu'à ce qu'elles soient nécessaires*

*Explication: L'évaluation paresseuse diffère le calcul des expressions jusqu'à ce qu'elles soient nécessaires, ce qui peut améliorer l'efficacité en évitant des calculs inutiles.*