

## Algo et Dev – QCM - Réponses

**Q1 : Quelle est la complexité moyenne d'accès à un élément dans une liste chaînée ?**

c)  $O(n)$

**Q2 : Dans une pile (stack), l'élément inséré en dernier est :**

a) Le premier à sortir

**Q3 : La différence principale entre une pile et une file est :**

b) L'ordre de retrait

**Q4 : Une file (queue) suit le principe :**

b) FIFO

**Q5 : Une liste doublement chaînée permet :**

c) De parcourir dans les deux sens

**Q6 : Quelle structure de données est la plus adaptée pour implémenter un système de backtracking ?**

a) Pile

## Programmation Orientée Objet (OOP)

**Q7 : Une classe définit :**

b) Un modèle d'objet

**Q8 : Le polymorphisme permet :**

a) D'utiliser plusieurs classes différentes avec une même interface

**Q9 : L'héritage multiple est :**

c) Autorisé selon le langage

**Q10 : Une méthode abstraite :**

b) N'a pas d'implémentation

**Q11 : Une interface sert à :**

- a) Définir un contrat sans implémentation

**Q12 : L'encapsulation permet de :**

- a) Cacher les détails internes d'une classe

**Q13 : Une méthode statique :**

- a) Peut être appelée sans instance

**Q14 : En Python, le mot-clé `super()` sert à :**

- a) Appeler la méthode parente

## Clean Architecture

**Q15 : Le principe fondamental de la Clean Architecture est :**

- a) Séparer les couches métier et infrastructure

**Q16 : La dépendance inversée signifie :**

- a) Le domaine ne dépend pas des détails techniques

**Q17 : Dans une architecture propre, les entités :**

- a) Ne doivent pas dépendre des frameworks

**Q18 : Le but des "use cases" est de :**

- a) Exprimer les règles métier

## FastAPI

**Q19 : FastAPI est principalement basé sur :**

- c) Starlette et Pydantic

**Q20 : Le décorateur `@app.get("/")` définit :**

- a) Une route GET

**Q21 : Pour définir un modèle de données dans FastAPI, on utilise :**

- c) BaseModel de Pydantic

**Q22 : FastAPI est connu pour :**

- a) Sa rapidité et la validation automatique

**Q23 : Pour lancer une application FastAPI, on utilise :**

- b) `uvicorn main:app --reload`

## LLM (Large Language Models)

**Q24 : Un LLM est principalement entraîné sur :**

- b) Du texte

**Q25 : Le but d'un LLM est de :**

- a) Prédire la prochaine séquence de mots

**Q26 : Les modèles comme GPT utilisent :**

- a) Des réseaux de neurones de type Transformer

**Q27 : Le prompt engineering consiste à :**

- a) Optimiser la formulation d'entrée

**Q28 : Les API comme OpenAI ou Ollama permettent :**

- a) D'intégrer un LLM dans une application