**QCM : Introduction à la Théorie de Compilation | Description de la phase d’un compilateur**

1. Quelle est la principale fonction d'un compilateur ?

- A) Exécuter le code source

- B) Traduire le code source en code machine

- C) Vérifier la grammaire du code source

- D) Optimiser le code source

2. Quel est le premier stade d'un compilateur ?

- A) L'analyse lexicale

- B) L'analyse syntaxique

- C) La génération de code

- D) L'optimisation

3. Qu'est-ce qu'un token en analyse lexicale ?

- A) Une unité syntaxique

- B) Une unité lexicale

- C) Une unité sémantique

- D) Une unité de traitement

4. Quel est le rôle de l'analyse syntaxique ?

- A) Vérifier la validité des tokens

- B) Générer le code machine

- C) Construire un arbre syntaxique

- D) Optimiser le code intermédiaire

5. Qu'est-ce qu'une grammaire context-free ?

- A) Une grammaire où chaque règle a une seule variable

- B) Une grammaire où les règles ne dépendent pas du contexte

- C) Une grammaire pour les langages naturels

- D) Une grammaire qui utilise des variables non définies

6. Quel est le but de la génération de code intermédiaire ?

- A) Produire un code directement exécutable

- B) Faciliter l'optimisation du code

- C) Traduire le code source en langage machine

- D) Vérifier la syntaxe du code

7. Quelle est la différence entre un compilateur et un interpréteur ?

- A) Le compilateur traduit le code en temps réel, l'interpréteur le traduit en une seule fois

- B) Le compilateur traduit le code source en une fois, l'interpréteur l'exécute ligne par ligne

- C) Le compilateur utilise des tokens, l'interpréteur des variables

- D) Le compilateur optimise le code, l'interpréteur ne le fait pas

8. Qu'est-ce qu'une table de symboles ?

- A) Une structure pour stocker des variables et des constantes

- B) Un outil pour vérifier les erreurs lexicales

- C) Une liste des erreurs de compilation

- D) Un code intermédiaire généré par le compilateur

9. Quel est le rôle de l'optimisation dans un compilateur ?

- A) Réduire le temps d'exécution et l'utilisation de la mémoire

- B) Vérifier la grammaire du code source

- C) Générer du code source depuis le code machine

- D) Vérifier les erreurs sémantiques

10. Quelle structure est généralement utilisée pour représenter la hiérarchie syntaxique d'un programme ?

- A) Un diagramme de flux

- B) Un arbre syntaxique

- C) Une table de symboles

- D) Un graphe de contrôle

11. Qu'est-ce qu'un 'parse tree' ?

- A) Un arbre représentant la structure lexicale du code source

- B) Un arbre représentant la structure syntaxique du code source

- C) Un arbre représentant les opérations de compilation

- D) Un arbre représentant le code machine généré

12. Quel est l'objectif de l'analyse sémantique ?

- A) Vérifier la conformité du code aux règles de syntaxe

- B) Assurer que les instructions ont un sens logique et correct

- C) Générer du code machine optimisé

- D) Détecter les erreurs de syntaxe

13. Quel type de langage de programmation est généralement associé à une grammaire context-free ?

- A) Langage de bas niveau

- B) Langage de haut niveau

- C) Langage de script

- D) Langage de machine

14. Quelle est la fonction principale d'un analyseur lexical (lexer) ?

- A) Vérifier la syntaxe du code source

- B) Générer un code intermédiaire

- C) Extraire les tokens du code source

- D) Optimiser le code machine

15. Que fait un 'code generator' dans un compilateur ?

- A) Convertit le code source en code intermédiaire

- B) Optimise le code intermédiaire

- C) Génère le code machine à partir du code intermédiaire

- D) Analyse la syntaxe du code source

16. Quel est le rôle du 'linker' dans le processus de compilation ?

- A) Compiler le code source

- B) Lier les modules du programme et résoudre les références externes

- C) Vérifier la validité du code intermédiaire

- D) Optimiser le code machine

17. Quelle est une caractéristique d'un langage compilé par rapport à un langage interprété ?

- A) Le langage compilé est généralement plus facile à déboguer

- B) Le langage compilé est souvent plus rapide à l'exécution

- C) Le langage compilé nécessite moins de ressources mémoire

- D) Le langage compilé est souvent plus simple à écrire

18. Qu'est-ce qu'une erreur de compilation ?

- A) Une erreur détectée pendant l'exécution du programme

- B) Une erreur liée à la syntaxe ou à la sémantique du code source, détectée lors de la compilation

- C) Une erreur dans le code machine généré

- D) Une erreur dans l'optimisation du code

19. Quel est l'objectif principal de l'analyse syntaxique ?

- A) Générer le code intermédiaire

- B) Vérifier la structure grammaticale du code source

- C) Optimiser le code source

- D) Assurer la validité sémantique du code

20. Quel est le rôle de l'analyse de dépendance dans l'optimisation du code ?

- A) Identifier les variables globales dans le code

- B) Détecter les erreurs de syntaxe

- C) Déterminer les dépendances entre instructions pour optimiser leur ordre d'exécution

- D) Traduire les variables locales en variables globales