



Théorie des langages et compilation  
Contrôle continu numéro 1  
(20 minutes)

Nom et prénom :

Malléjac Clément

Noircissez les bonnes réponses (cocher ne suffit pas). Les questions faisant apparaître le symbole ♣ peuvent présenter une ou plusieurs bonnes réponses ; les autres ont une seule bonne réponse. Toute absence de réponse équivaut à une réponse fausse. Utilisez le verso des feuilles comme brouillon si nécessaire.

Rôle d'un compilateur

Question 1 ♣ Quel est la traduction anglaise de compilateur ?

☐ parser  
☐ compiler

☐ word processor  
☐ assembler

☒ language processor  
☒ compiler

Question 2 ♣ Parmi les langages suivants, lesquels sont de bas niveau ?

☐ le langage C  
☐ le shell Unix

☒ le code machine  
☒ le langage d'assemblage

☒ un langage dans lequel chaque instruction correspond à une instruction machine

Question 3 ♣ Parmi les langages suivants, lesquels sont de haut niveau ?

☒ Fortran  
☒ C  
☒ Basic

☒ le shell Unix  
☒ Python  
☐ le langage d'assemblage

Question 4 Un assembleur est un interpréteur.

☒ faux

☐ vrai

Question 5 Un processeur est un traducteur.

☒ faux

☐ vrai

Question 6 Un compilateur traduit chaque instruction du programme source par une seule instruction machine.

☒ faux

☐ vrai

Question 7 Un langage compilé est traduit en code machine par un compilateur

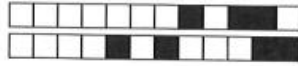
☒ vrai

☐ faux

Question 8 Un interpréteur traduit la totalité du programme source avant de commencer à l'exécuter.

☒ faux

☐ vrai



**Question 9** Un compilateur traduit un programme source vers un langage cible exécutable.

☐ faux

☒ vrai

**Question 10** Une machine virtuelle est un interpréteur.

☐ faux

☒ vrai

**Question 11** Le programme source d'un langage semi-interprété est traduit partiellement avant d'être interprété.

☒ faux

☐ vrai

**Question 12** Le programme source d'un langage compilé est traduit complètement avant d'être exécuté.

☐ faux

☒ vrai

**Question 13** Le pré-traitement effectué par un interpréteur est plus important que celui réalisé par un compilateur.

☐ vrai

☒ faux

**Question 14** La compilation à la volée (JIT) dans une JVM consiste à traduire du bytecode Java en code binaire à l'exécution.

☒ vrai

☐ faux

**Question 15** Les langages compilés s'exécutent plus vite que les langages interprétés.

☐ faux

☒ vrai

**Question 16** Les langages semi-interprétés s'exécutent plus vite que les langages compilés.

☒ faux

☐ vrai

**Question 17 ♣** On attend d'un compilateur :

☐ qu'il rejette les programmes contenant trop de variables (> 1000)

☐ qu'il exécute le programme s'il ne comporte pas d'erreur

☒ qu'il produise du code efficace

☐ qu'il corrige les erreurs de programmation

☒ qu'il produise du code compact

**Question 18 ♣** Parmi les outils suivants, lesquels sont des programmes ?

☐ Une mémoire virtuelle

☒ Un assembleur

☒ Un éditeur de liens

☒ Un compilateur

☒ Une machine virtuelle

☒ Un interpréteur

### Etape d'une compilation

**Question 19** Un programme source écrit dans un langage de programmation est un fichier au format texte.

☒ vrai

☐ faux



Question 20 L'analyse lexicale produit un arbre.

☒ faux

☐ vrai

Question 21 L'analyse syntaxique produit un arbre et une table.

☒ vrai

☐ faux

Question 22 AST est l'acronyme de Arbre Syntaxique Translaté

☐ vrai

☒ faux

Question 23 L'analyse sémantique corrige les erreurs de type.

☒ faux

☐ vrai

Question 24 L'analyseur sémantique produit du code intermédiaire.

☐ vrai

☒ faux

Question 25 Le générateur de code intermédiaire prend en entrée un arbre.

☐ faux

☒ vrai

### Méthode de conception

Question 26 Le paradigme analyse-synthèse est un découpage fonctionnel du compilateur.

☐ faux

☒ vrai

Question 27 Les langages formels sont plus expressifs que les langages naturels.

☐ vrai

☒ faux

Question 28 La Théorie des Langages a permis de rationaliser la conception des compilateur.

☒ vrai

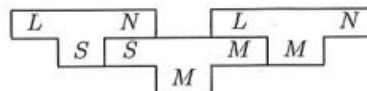
☐ faux

Question 29 Le formalisme BNF permet de décrire la grammaire d'un langage de programmation.

☒ vrai

☐ faux

Question 30 Le schéma suivant illustre une compilation croisée de  $L$  pour  $N$  sur  $M$ .



☐ faux

☒ vrai

Question 31 La compilation croisée permet de produire un compilateur qui s'exécute sur une autre machine.

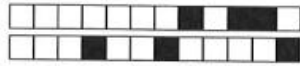
☒ vrai

☒ faux

Question 32 Le premier compilateur a été écrit pour le langage Fortran.

☐ faux

☒ vrai

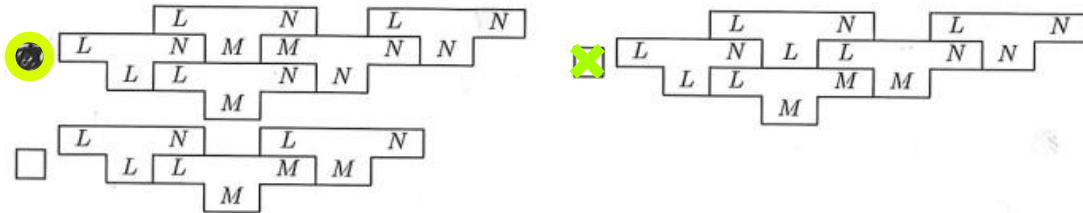


**Question 33** L'auto-compilation permet d'écrire un compilateur dans le langage qu'il compile.

☒ vrai

☐ faux

**Question 34** On dispose d'un compilateur d'un langage  $L$  écrit en  $L$  produisant du code pour une machine  $M$ . Et on dispose du compilateur du langage  $L$  sur  $M$ . On réécrit le backend du compilateur de  $L$  pour une machine  $N$  et on obtient un compilateur de  $L$  sur une machine  $N$  de la manière suivante :



**Question 35** La partie du compilateur avant réalise l'analyse du programme source.

☒ vrai

☐ faux

**Question 36** Le back-end du compilateur produit le programme cible.

☐ faux

☒ vrai

**Question 37** Le paradigme analyse-synthèse facilite l'exécution du compilateur.

☒ vrai

☒ faux

**Question 38** Pour recibler un compilateur il faut réécrire seulement sa partie arrière.

☐ faux

☒ vrai

**Question 39** ♣ Quelles étapes contient la partie avant du compilateur (l'ordre donné ci-dessous ne doit pas être pris en compte) ?

☒ l'analyse syntaxique

☐ l'optimisation sémantique

☒ l'analyse lexicale

☒ la génération de code intermédiaire

**Question 40** ♣ Quelles étapes contient la partie arrière du compilateur (l'ordre donné ci-dessous ne doit pas être pris en compte) ?

☒ optimisation du code cible

☐ l'analyse syntaxique

☒ la génération de code cible

☐ l'analyse lexicale

**Question 41** ♣ Dans quelle partie est utilisée l'arbre syntaxique abstrait ?

☒ Dans la partie avant

☐ Dans la partie arrière