

Il y a cinq exercices, notés sur 5 points chacun. Il est donc possible **d'en choisir seulement 4.**

Exercice 1 : Questions de cours

1. À quoi servent les types dans un langage de programmation ?
2. Quels sont les avantages d'un langage compilé par rapport à un langage interprété ?
3. Java est-il un langage compilé ou interprété ?
4. Quelle est la différence entre le langage MIPS32 et le langage de la représentation intermédiaire vue en cours ?
5. Peut-on évaluer toutes les expressions du langage lors de la compilation ? Justifiez.

Exercice 2 : Langages

question 1 : Écrivez un fichier Antlr4 pour générer un analyseur lexical et syntaxique reconnaissant un langage comportant :

- Les expressions : variables, entiers, sommes, produit, appels de fonctions.
- Les instructions : conditionnelle, boucle, affectation, déclaration de fonction.

question 2 : Dérivation. Considérez la grammaire suivante, dont les terminaux sont a, b , et c :

$S := EaG|Ea$

$E := b|FaF$

$F := Ec|\epsilon$

$G := Fc|S$

1. Cette grammaire est-elle ambiguë ? Justifiez.
2. Quels sont les non-terminaux ?
3. $cbcaacba$ est-il un mot de cette grammaire ? Justifiez
4. Même question pour $bcaac$.

Exercice 3 : Analyse - Graphes

Considérez l'extrait de programme suivant :

```
1      int m=0;
      int n=0;
3      int y=n;
      print(m);
5      while(y<10){
          int z=2;
7          int w=y+z;
      }
9      int x=w;
      return x;
11     print(x);
```

1. Dessinez son graphe de flot de contrôle.

2. Comment peut-on l'utiliser pour optimiser le code (en général, et dans cet exemple en particulier) ?
3. Peut-on détecter (toutes) les boucles infinies lors de la compilation d'un programme ? Si oui, comment, si non, pourquoi ?
4. Dessinez le graphe d'interférence (Bonus : avec les arrêtes de préférence).
5. À quoi sert ce graphe d'interférence ? Quel algorithme lui appliquera-t-on ? Que pourra-t-on en conclure pour le cas de cet exemple ?

Exercice 4 : Frames et assembleur

Considérez la fonction suivante :

```
1      int f(int x, int y, int z){
2          int w = x+y+z;
3          int i = g(w);
4          return i;
5      }
```

Où on suppose que g est une fonction bien typée dans le contexte, définie précédemment dans le code.

1. Lors de la traduction de ce frame en assembleur, quelle taille devra lui être réservée sur la pile lors de la mise à jour de $\$fp$ et de $\$sp$? On rappelle que les entiers et les adresses sont codés sur 4 octets. Justifiez précisément.
2. Décrivez le code assembleur généré par la compilation du code ci-dessus. Votre description pourra décrire l'algorithme de manière informelle (sans la syntaxe précise de MIPS32), et être accompagnée de schémas.

Exercice 5 : Résumé

Que fait un compilateur ? Décrivez précisément les différentes phases, avec leurs responsabilités et leurs utilités respectives. Mentionnez les optimisations.