

Votre nom : \_\_\_\_\_

## CMP2 – Construction des compilateurs

EPITA – Ing1 promotion 2014

**Tous documents (notes de cours, photocopiés, livres) autorisés  
Calculatrices, ordinateurs, tablettes et téléphones interdits.**

Juin 2012 (1h30)

Lisez bien les questions, chaque mot est important. Écrivez court, juste et bien ; servez vous d'un brouillon. Une argumentation informelle mais convaincante est souvent suffisante. Gérez votre temps, ne restez pas bloqué sur les questions les plus difficiles. Une lecture préalable du sujet est recommandée.

Ce sujet contient 8 pages. Les pages 1–5 contiennent l'épreuve elle-même. Ce document comporte en pages 6–8 une enquête (facultative) sur le cours et le projet Tiger (y répondre sur la feuille de QCM qui vous est fournie).

**Écrivez votre nom en haut de la première page du sujet et rendez-le avec votre copie et la feuille de QCM.**

### 1 La mémoire dans tous ses états

1. En compilation, qu'appelle-t-on une activation ?
2. Qu'est-ce qu'un bloc d'activation ? Que contient-il ?
3. Citez un mécanisme d'allocation dynamique sur la pile en C et un mécanisme d'allocation dynamique sur le tas en C.
4. Quelle différence faites-vous entre l'allocation dynamique sur la pile (*stack*) et l'allocation dynamique sur le tas (*heap*) ?
5. Comment fonctionne la libération de la mémoire dans chaque cas (allocation dynamique sur la pile, allocation dynamique sur le tas) ?
6. Quels sont les avantages de chaque système ?
7. Rappeler quelles sont les stratégies d'allocations utilisées en Tiger et pour quels types de données elles sont utilisées.
8. Soit la fonction Tiger *f* suivante :

```
function f (x : int) : int = x + 1
```

tc génère pour *f* le code MIPS suivant :

```

1 tc_l0:
2     sw    $fp, -8 ($sp)
3     move  $fp, $sp
4     sub   $sp, $sp, 12
5     sw    $a0, ($fp)
6     sw    $a1, -4 ($fp)
7 l1:
8     lw    $t0, -4 ($fp)
9     add   $v0, $t0, 1
10 l2:
11     move  $sp, $fp
12     lw    $fp, -8 ($fp)
13     jr    $ra
    
```

- (a) Quelle est la taille du cadre de pile de f ?
- (b) Détaillez le contenu de ce cadre de pile.
- (c) tc pourrait tout à fait générer le code de f sans utiliser le frame pointer (\$fp). Réécrivez ce code sans vous servir de \$fp.
- (d) La suppression du frame pointer du code généré, illustrée dans la question précédente, est en fait une optimisation proposée par certains compilateurs, qui permet de simplifier les prologues et les épilogues des routines. Il existe cependant des situations dans lesquelles cette optimisation n'est pas envisageable. Citez-en une.

## 2 Partie terminale

Dans cet exercice, on considère une machine hypothétique, dont le jeu d'instructions est suffisamment clair pour ne pas avoir à être défini et qui possède trois registres r1, r2 et r3. Les deux premiers sont sous la responsabilité de l'appelant (*caller-save*) et le dernier sous celle de l'appelé (*callee-save*). Par soucis de simplification, on considèrera comme complètement implicite la gestion d'éléments tels que l'adresse de retour d'une routine, le frame pointer ou le stack pointer.

1. Définissez la partie terminale d'un compilateur.
2. Considérons le programme suivant, constitué de deux fonctions f et g, pour lequel l'allocation des registres n'a pas encore été effectuée.

<pre> 1 f:    t ← r3 2      n ← r1 3      r1 ← n 4      r2 ← 1 5      call g 6      v ← r1 7      r1 ← v 8      r3 ← t 9      return     </pre>	<pre> 10 g:   t ← r3 11     a ← r1 12     b ← r2 13     if a &lt;= 1 jump l1 14     d ← b * a 15     c ← a - 1 16     r1 ← c 17     r2 ← d 18     call g 19     v ← r1 20     jump exit 21 l1:  v ← b 22 exit: r1 ← v 23     r3 ← t 24     return     </pre>
---	--

Écrivez un programme Tiger ou C équivalent à ce programme.

3. À quoi peut servir ce programme ?
4. D'après ce programme, détailler les conventions d'appel de cette machine ; en particulier, pour chacun des registres, dire lesquels servent au passage d'argument(s) et au retour de résultat.
5. Étant données les conventions d'appel, le sens du programme, expliquer quelles sont les variables vives lors du return de la fonction g (ligne 24).
6. Expliquez pourquoi les lignes

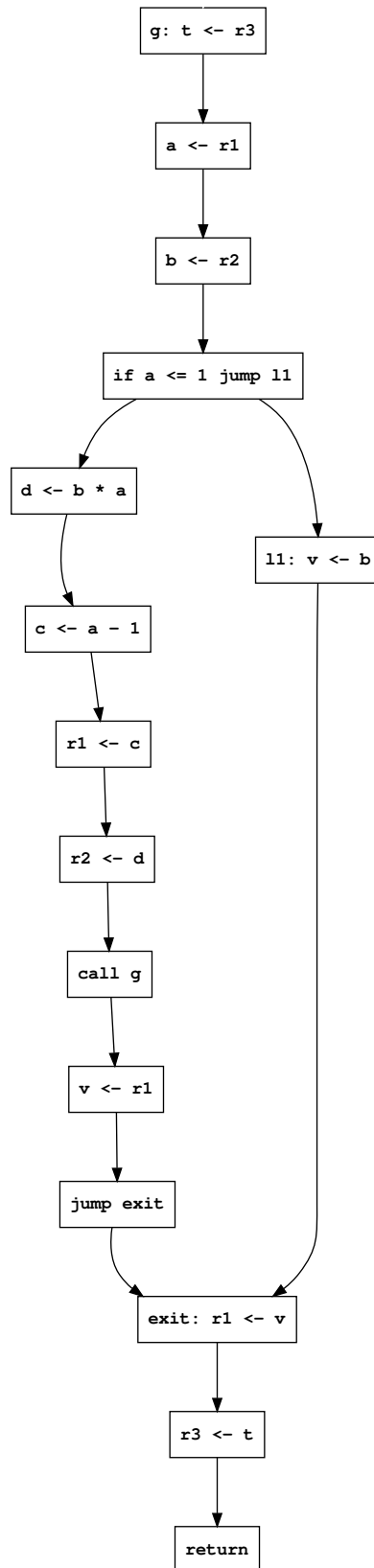
```
t ← r3
```

et

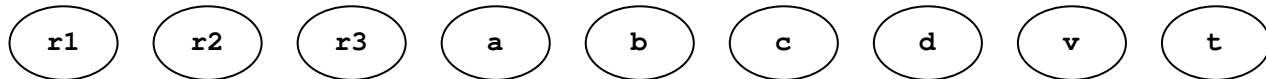
```
r3 ← t
```

ont été générées.

7. Étiquetez les arcs du graphe de contrôle de flux de la fonction g ci-après avec les temporaires vives. N'oubliez pas d'indiquer également les temporaires vives en entrée et en sortie du programme. **Répondez directement sur le sujet.**



8. Complétez le graphe d'interférence ci-dessous en traçant une arête pour chaque interférence entre deux variables. Faites-y aussi figurer les fusions possibles (*coalesces*) avec des arêtes en pointillés (et de préférence en utilisant un stylo d'une autre couleur). **Répondez directement sur le sujet.**



9. Le graphe précédent est-il colorable ?
- Si oui, précisez quelles registres vous affectez à chaque variable (de préférence en coloriant directement le graphe précédent en utilisant des stylos de couleurs **différentes** ou à défaut en indiquant ces affectations sur votre copie).
  - Si non, indiquez et justifiez le meilleur choix de variable à verser sur la pile, puis réécrivez le code de la fonction *g* en intégrant le versement de cette variable sur la pile.
10. **Bonus.** Le code de la fonction *g*, visible à la question 2, est un cas de récursion terminale (*tail-recursion*) : *g* s'appelle récursivement (cf. ligne 18), mais cet appel est la dernière instruction à être évaluée (avant le retour de la fonction). Dans cette situation, il est possible d'optimiser le code en transformant cette récursion en une *itération*. Réécrivez le code de *g* en lui appliquant cette optimisation.

### 3 À propos de ce cours

Pour terminer cette épreuve, nous vous invitons à répondre à un petit questionnaire. Les renseignements ci-dessous ne seront bien entendu pas utilisés pour noter votre copie. Ils ne sont pas anonymes, car nous souhaitons pouvoir confronter réponses et notes. En échange, quelques points seront attribués pour avoir répondu. Merci d'avance.

Sauf indication contraire, vous pouvez cocher plusieurs réponses par question. Répondez sur la feuille de QCM. N'y passez pas plus de dix minutes.

#### Cette épreuve

Q.1 Sans compter le temps mis pour remplir ce questionnaire, combien de temps ce partiel vous a-t-il demandé (si vous avez terminé dans les temps), ou combien vous aurait-il demandé (si vous aviez eu un peu plus de temps pour terminer) ?

- a. Moins de 30 minutes.
- b. Entre 30 et 60 minutes.
- c. Entre 60 et 90 minutes.
- d. Entre 90 et 120 minutes.
- e. Plus de 120 minutes.

Q.2 Ce partiel vous a paru

- a. Trop difficile.
- b. Assez difficile.
- c. D'une difficulté normale.
- d. Assez facile.
- e. Trop facile.

#### Le cours

Q.3 Quelle a été votre implication dans les cours CMP2 ?

- a. Rien.
- b. Bachotage récent.
- c. Relu les notes entre chaque cours.
- d. Fait les annales.
- e. Lu d'autres sources.

Q.4 Ce cours

- a. Est incompréhensible et j'ai rapidement abandonné.
- b. Est difficile à suivre mais j'essaie.
- c. Est facile à suivre une fois qu'on a compris le truc.
- d. Est trop élémentaire.

Q.5 Ce cours

- a. Ne m'a donné aucune satisfaction.
- b. N'a aucun intérêt dans ma formation.
- c. Est une agréable curiosité.
- d. Est nécessaire mais pas intéressant.
- e. Je le recommande.

Q.6 La charge générale du cours en sus de la présence en amphi (relecture de notes, compréhension, recherches supplémentaires, etc.) est

- a. Telle que je n'ai pas pu suivre du tout.
- b. Lourde (plusieurs heures de travail par semaine).
- c. Supportable (environ une heure par semaine).
- d. Légère (quelques minutes par semaine).

#### Les formateurs

Q.7 L'enseignant

- a. N'est pas pédagogue.
- b. Parle à des étudiants qui sont au dessus de mon niveau.
- c. Me parle.
- d. Se répète vraiment trop.
- e. Se contente de trop simple et devrait pousser le niveau vers le haut.

Q.8 Les assistants

- a. Ne sont pas pédagogues.

- b. Parlent à des étudiants qui sont au dessus de mon niveau.
- c. M'ont aidé à avancer dans le projet.
- d. Ont résolu certains de mes gros problèmes, mais ne m'ont pas expliqué comment ils avaient fait.
- e. Pourraient viser plus haut et enseigner des notions supplémentaires.

### Le projet Tiger

- Q.9 Vous avez contribué au développement du compilateur de votre groupe (une seule réponse attendue) :
- a. Presque jamais.
  - b. Moins que les autres.
  - c. Équitablement avec vos pairs.
  - d. Plus que les autres.
  - e. Pratiquement seul.
- Q.10 La charge générale du projet Tiger est
- a. Telle que je n'ai pas pu suivre du tout.
  - b. Lourde (plusieurs jours de travail par semaine).
  - c. Supportable (plusieurs heures par semaine).
  - d. Légère (une ou deux heures par semaine).
  - e. J'ai été dispensé du projet.
- Q.11 Y a-t-il de la triche dans le projet Tiger ? (Une seule réponse attendue.)
- a. Pas à votre connaissance.
  - b. Vous connaissez un ou deux groupes concernés.
  - c. Quelques groupes.
  - d. Dans la plupart des groupes.
  - e. Dans tous les groupes.

**Questions 12-18** Le projet Tiger vous a-t-il bien formé aux sujets suivants ? Répondre selon la grille qui suit. (Une seule réponse attendue par question.)

- |                      |                   |                       |               |                    |
|----------------------|-------------------|-----------------------|---------------|--------------------|
| <b>a</b> Pas du tout | <b>b</b> Trop peu | <b>c</b> Correctement | <b>d</b> Bien | <b>e</b> Très bien |
|----------------------|-------------------|-----------------------|---------------|--------------------|

- Q.12 C++.
- Q.13 modélisation orientée objet et *design patterns*.
- Q.14 anglais technique.
- Q.15 compréhension du fonctionnement des ordinateurs.
- Q.16 compréhension du fonctionnement des langages de programmation.
- Q.17 travail collaboratif.
- Q.18 outils de développement (contrôle de version, systèmes de construction, débogueurs, générateurs de code, etc.)

**Questions 19-34** Comment furent les étapes du projet ? Répondre selon la grille suivante. (Une seule réponse attendue par question ; ne pas répondre pour les étapes que vous n'avez pas faites.)

- |                       |                  |                  |                     |                          |
|-----------------------|------------------|------------------|---------------------|--------------------------|
| <b>a</b> Trop facile. | <b>b</b> Facile. | <b>c</b> Nickel. | <b>d</b> Difficile. | <b>e</b> Trop difficile. |
|-----------------------|------------------|------------------|---------------------|--------------------------|

- Q.19 Rush .tig : mini-projet en Tiger (Logomatig).
- Q.20 TC-0, Scanner & Parser.
- Q.21 TC-1, Scanner & Parser, Tâches, Autotools.
- Q.22 TC-2, Construction de l'AST et pretty-printer.

- Q.23 TC-3, Liaison des noms et renommage.
- Q.24 TC-4, Typage et calcul des échappements.
- Q.25 TC-5, Traduction vers représentation intermédiaire.
- Q.26 TC-6, Simplification de la représentation intermédiaire.
- Q.27 TC-7, Sélection des instructions.
- Q.28 TC-8, Analyse du flot de contrôle.
- Q.29 TC-9, Allocation de registres.
- Q.30 TC-D (option), Désucrage (boucles for, comparaisons de chaînes de caractères).
- Q.31 TC-I (option), Mise en ligne du corps des fonctions.
- Q.32 TC-B (option), Vérification dynamique des bornes de tableaux.
- Q.33 TC-A (option), Surcharge des fonctions.
- Q.34 TC-0 (option), Désucrage des constructions objets.