



Concours National d'Informatique
Sujet de demi-finale Lyon / Bordeaux

13 Février 2010

C'EST MON CHOIX !

1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre demi-finale. Sa durée est de 3 heures. Par la suite, vous passerez un entretien (20 minutes) et une épreuve de programmation sur machine (4 heures).

Conseils

- Lisez bien tout le sujet avant de commencer.
- **Soignez la présentation** de votre copie.
- N'hésitez pas à poser des questions.
- Si vous avez fini en avance, relisez bien.
- N'oubliez pas de passer une bonne journée.

Remarques

- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez la demi-finale et la date, en haut de votre copie.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le nous, nous pouvons ajouter des questions plus difficiles.
- Tous les langages sont autorisés. Néanmoins, veuillez préciser celui que vous utilisez.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe.

2 Sujet

Introduction

Une fois de plus Joseph Marchand fait appel aux jeunes virtuoses de la ligne de code de Prolog afin de l'aider dans sa vie quotidienne. En effet en tant qu'estète de l'octet il cherche à optimiser les choix qu'il est amené à prendre au quotidien. Que manger ? Où aller ? Que faire ? Quelle est la réponse à la grande question sur la Vie, l'Univers et le Reste ?

Formalisons...

Joseph Marchand étant plutôt binaire, ses choix le sont également. Afin de les optimiser il les structure en formules composées des opérateurs booléens habituels (non \neg , ou inclusif \vee , et \wedge). Une formule représente donc un ensemble de choix booléens et s'évalue en un résultat booléen. Par exemple, si Joseph Marchand a faim il pourra, A manger une pomme ou B manger une orange. Soit : $F = A \vee B$

Le but du jeu va être d'aider Joseph Marchand à savoir quels choix prendre pour satisfaire ses besoins.

Question 0 (42 points)

Combien de valeurs différentes peut prendre un booléen sur une architecture 64 bits ?

Question 1 (2 points)

Écrivez une structure de donnée qui permet de stocker une formule.

Décrivez très précisément quelles sont les propriétés de votre structure et comment elle s'utilise.

Représentez graphiquement votre structure de donnée pour les formules suivantes :

1. $F = A$
2. $F = A \vee B$
3. $F = A \wedge B$
4. $F = A \wedge (B \vee C)$
5. $F = (A \wedge B) \vee (A \wedge C)$

Remarque 1 : La plus grande difficulté de ce sujet est de déterminer la bonne structure. Veillez à bien lire tout le sujet plusieurs fois afin de voir comment vous allez devoir utiliser votre structure. Soyez innovants !

Remarque 2 : Dans le reste du sujet vous allez devoir manipuler des tableaux de booléens indexés par un nom de variable. Le nom d'une variable est toujours composé d'un seul caractère alphabétique majuscule. Le détail d'implémentation du tableau est laissé à votre discrétion.

Question 2 (3 points)

Écrivez une fonction qui évalue une formule étant donné une valeur pour chacune de ses variables.

Prototype

En entrée : votre structure représentant une formule et un tableau associant pour chaque variable une valeur booléenne.

En sortie : une valeur booléenne.

Exemple

En entrée : $F = (A \vee B) \wedge C$, $[A = \text{vrai}, B = \text{faux}, C = \text{vrai}]$

En sortie : vrai

Question 3 (3 points)

Écrivez une fonction qui détermine si une formule peut être vraie.

Prototype

En entrée : votre structure représentant une formule.

En sortie : une valeur booléenne.

Exemple

En entrée : $F = (A \wedge B) \wedge \neg A$

En sortie : faux

Question 4 (4 points)

Écrivez une fonction qui prend une formule et la valeur d'une de ses variables et qui détermine s'il est possible que la formule soit vraie étant donné la valeur de la variable.

Prototype

En entrée : votre structure et une valeur de variable

En sortie : une valeur booléenne

Par exemple

En entrée : $F = (A \wedge B) \vee C$, $C = \text{faux}$

En sortie : vrai

Question 5 (6 points)

Écrivez une fonction qui prend une formule et qui renvoie une liste de valeurs de variables telle que la formule soit vraie.

Prototype

En entrée : votre structure représentant une formule. En sortie : une liste de valeurs booléennes.

Par exemple

En entrée : $F = (A \vee B) \wedge C$

En sortie : $[A = \text{faux}, B = \text{vrai}, C = \text{vrai}]$

Remarque

On vous demande *une* liste, sous-entendu qu'il y a plusieurs réponses possibles. Toutes sont valables, la liste n'a pas besoin d'être optimale, la seule contrainte est que la fonction écrite en 2 doit renvoyer vrai si on lui passe la formule et la liste résultat.

Question 6 (2 points)

Joseph Marchand aimerait emporter votre programme partout avec lui. Pour ce faire, il vous demande donc d'estimer le temps que prendrait votre algorithme de la question 5 sachant qu'il utilise un Nokia N900 disposant d'un processeur ARM à 600 Mhz.

Le téléphone ne dispose que de 768 Mo de mémoire vive, est-ce que cela risque de poser un problème ? Calculez approximativement la consommation mémoire de votre structure de données pour une formule à N variables et M opérations. Quelles optimisations pourriez-vous faire pour diminuer la taille de votre structure ?

Si vous avez terminé prenez le temps de relire et vérifier vos réponses et de rajouter quelques commentaires. Si vous avez trop de temps voici une question bonus :

Bonus – Question 7

Écrivez une fonction qui compose deux formules par un opérateur binaire.

Prototype

En entrée : deux formules représentées par votre structure et un opérateur binaire (donné sous la forme qui vous arrange).

En sortie : votre structure représentant une formule.

Par exemple

En entrée : $F = A \wedge B, G = (A \vee B) \wedge C, \wedge$

En sortie : $F = A \wedge B \wedge C$