Licence L2 MI 2014 – 2015
Module *Principes d'Interprétation des Langages* Feuille de TP n° 4

Contrôle de connaissances

Les réponses doivent être envoyées à votre chargé de TP par mail avant la fin de la séance. Le mail doit contenir les fichiers Lex et Yacc (avec une extension ".l" et ".y") avec des noms représentatifs (e.g., prénom_nom_exo1.l). Pour être évalués, tous les fichiers doivent pouvoir être compilés sans erreurs.

`elena.knyazeva@limsi.fr`

I) Exercice 1 : Lex.

L'objectif de cet exercice est d'extraire des informations simples d'un fichier de format spécifié. Le fichier représente le résultat des élections qui ont eu le 17 avril de 9h00 à 18h00. Chaque ligne contient 4 champs séparés par des espaces : le nom de la personne votant, l'heure du premier vote, l'heure de la dernière modification (qui est égale à l'heure du premier vote, s'il n'y a eu aucune modification), le vote lui-même ("pour" ou "contre"). Il peut y avoir des lignes commençant par # qui sont des commentaires et qui doivent être ignorées. Voici un exemple :

```
# c'est un exemple --------------
Martin 09:38:49 09:38:49 pour
Dubois 16:30:45 17:08:10 contre
Bernard 12:49:08 12:49:08 pour
# ------------------------------
```

Vous devez répondre aux questions suivantes :

1. Combien y a-t-il de votes au total ?
2. Combien y a-t-il de "pour" et de "contre" ?
3. Qui a donné son premier vote entre 12h30 et 13h30 ?
4. Combien de personnes ont modifié leur vote au moins une fois ?

Vous trouverez un exemple de fichier de test à l'adresse suivante : `https://perso.limsi.fr/knyazeva/votes.txt`. Votre programme, une fois exécuté avec un fichier en entrée, doit afficher les réponses aux questions dans un format lisible. Le fichier utilisé pour évaluer vos résultats peut être différent du fichier fourni.

Indication : vous pouvez afficher les réponses dans un ordre arbitraire. Notamment, vous pouvez commencer par la question 3 pour afficher les noms des personnes demandées à la volée sans avoir à les stocker. Voici un exemple de sortie acceptable pour votre programme ;

```
Les personnes qui ont voté entre 12h30 et 13h30 sont : Bernard Leroy Durand
Il y a 15 votes au total
Il y a 8 votes ‘‘pour’’ et 7 votes ‘‘contre’’
4 participants ont modifié leur vote
```

II) Exercice 2 : Yacc.

1. L'objectif de cet exercice est de réaliser une calculatrice qui accepte la définition de variables dans des commandes séparées. Vous pouvez partir de la calculatrice simple (voir le sujet de TP Yacc `https://perso.limsi.fr/knyazeva/tp3.pdf`).

Votre programme doit accepter plusieurs commandes séparées par le symbole `\n`. Les commandes peuvent être de deux types : déclarations des variables et calcules simples. Voici un exemple d'utilisation :

```
> 2 + 3
Résultat : 5
> a = 2
a = 2 enrégistré
> a * 3
Résultat : 6
```

Toutes les variables seront stockées dans un tableau d'entiers. On se limite à 26 variables avec les noms a - z (en déduire une manière simple de passer d'un nom de variable à son indice dans le tableau).

2. Ajouter les opérations logiques : `/\` (conjonction), `\/` (disjonction), `->` (implication), `~` (négation). Toutes ces opérations sont associatives à gauches et leurs priorités ainsi que celles des opérations arithmétiques sont résumées dans le tableaux suivant.

1	~
2	*, /
3	+, -
4	/\
5	\/
6	->

Comme en C, la valeur logique faux est représentée par l'entier 0 et la valeur logique vrai par tout entier non nul. Le résultat des opérations logiques sera toujours représenté pars les valeurs 0 et 1.

Par exemple :

```
> p = 1
p = 1 enregistré
> q = 0
q = 0 enregistré
> r = 3
r = 3 enregistré
> p -> q
Résultat : 0
> q \/ r + 2
Résultat : 1
```

```
> (q \/ r) + 2
Résultat : 3
> q \/ r /\ q
Résultat : 0
> r \/ p -> q
Résultat : 0
> r \/ (p -> q)
Résultat : 1
```