

Lex, Yacc, XML, XSLT et AJAX

Utilisation de Lex & YACC

1 – Le responsable d'une formation doit s'assurer que l'ensemble des heures prévues dans le programme sont assurées.

Pour ce faire, il dispose d'un fichier qui se compose de deux parties :

- le programme du semestre avec la liste des UE. Pour chaque UE le nombre d'heures de CM, TD et TP à prévoir
- la liste des enseignants le nombre d'heures qui réaliseront (CM, TD et TP).

Voici un exemple de fichier :

```
UE 1 : CM : 9h, TD : 15h, TP : 30h
UE 2 : CM : 9h, TD : 15h, TP : 30h
UE 3 : CM : 6h, TD : 4.5h, TP : 9h
UE 4 : CM : 12h, TD : 15h, TP : 30h
-----
*Enseignant AB*
UE 1 : CM : 9h, TD : 15h, TP : 0h
UE 2 : CM : 9h, TD : 0h, TP : 0h
UE 3 : CM : 9h, TD : 0h, TP : 0h
;
*Enseignant CD*
UE 4 : CM : 12h, TD : 15h, TP : 30h
;
*Vacataire EF*
UE 2 : CM : 0h, TD : 15h, TP : 30h
;
*Vacataire EF*
UE 3 : CM : 0h, TD : 4.5h, TP : 4.5h
;
```

On remarque que :

- ▷ les UE sont identifiées par un identifiant ;
- ▷ pour chaque UE on précise le nombre d'heures dans l'ordre CM, TD, TP, même s'il n'y a pas d'heure prévue.

Questions :

- a. Écrivez un analyseur **lexical** et **syntactique** permettant de vérifier qu'un fichier est **conforme**.
- b. Modifier votre analyseur syntaxique pour qu'il affiche pour chaque enseignant, le nombre total d'heures équivalent TD qu'il devra réaliser.
(1h de CM = 1,5h équivalent TD, 1h de TD = 1h équivalent TD, 1h de TD = 0,5h équivalent TP).
- c. Modifier votre analyseur syntaxique pour qu'il vérifie s'il reste des heures à pourvoir ou si trop d'heures ont été programmées, pour chaque UE pour les CM, TD et TP.

Voici un exemple de sortie :

```
□ — xterm —
Résultat :
UE 1 : TP : 30.0h manquantes
UE 3 : CM : 3.0h en trop
UE 3 : TP : 4.5h manquantes
```

Vous fournirez un fichier **lex** et un fichier **yacc** permettant de répondre aux 3 questions.

■ ■ ■ Utilisation d'XML, XSLT et AJAX

2 – Vous récupérez la liste des programmes TV de la TNT sur https://xmltvfr.fr/xmltv/xmltv_tnt.xml

Le travail consiste à :

- ▷ **générer** un fichier XML faisant apparaître seulement toutes les "séries" de **toutes les chaînes** avec XSL.
Ce fichier sera lui même au format XML.
- ▷ Ce fichier est **consultable** au travers d'une page html avec Ajax qui devra l'afficher sous forme de tableau avec les icônes des chaînes.
Vous pourrez afficher dans un cadre le descriptif de l'épisode lorsque l'utilisateur clique sur une programmation.
Votre tableau peut être organisé par jour ou par chaîne pour tous les jours.
- ▷ on voudrait implémenter la possibilité de **filtrer par chaîne** :
Il faudra proposer à l'utilisateur de saisir le nom de la chaîne dans un champ puis de n'afficher que le programme de la chaîne choisie.

```
xterm
$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 ...
```

et vous connecter dessus à l'aide de l'URL : <http://localhost:8000/>.