

Mini-Projet

À déposer avant le 2 mai 2025 minuit, aucun délai ne sera accordé.

Déposer une archive `NOM1_NOM2.zip` sous format zip contenant les fichiers demandés. Vous déposerez également dans cette archive un fichier `README.txt` contenant le nom des binômes.

Remarques préliminaires :

- Il y a plusieurs questions, pour chacune, vous devrez déposer sur Updago les fichiers demandés.
- Attention, nous ne serons là pour vous aider et conseiller que pour les 2 premières heures, ne restez pas bloqués plusieurs heures sur une erreur de compilation sans avancer.
- L'évaluation tiendra beaucoup compte de la propreté de vos spécifications :
 - si vous avez bien mis dans la spécification `lex` ce qui doit être du traitement lexical, et dans la spécification `yacc` tout ce qui est analyse et traitement de la syntaxe du document ;
 - si vos spécifications `lex` et `yacc` sont claires et lisibles ;
 - si vos expressions rationnelles et votre grammaire traitent les choses simplement, clairement et proprement ;
 - si votre grammaire est telle que l'arbre de dérivation « reflète bien » la structure du document ;
 - tout en respectant toutes les contraintes du sujet.

Si tout cela est fait proprement et même si vos spécifications ne compilaient pas ou ne s'exécutaient pas correctement, elles seront bien plus appréciées que d'autres qui compileraient mais où tout serait fait en dépit du bon sens.

Analyse de billet de concert

Nous souhaitons traiter automatiquement des commandes de billets de concert, vous trouverez cet exemple `commandeBillet.txt` sur UPdago.

```
Dossier 12345678
ANNE-CLAIRE/BERGEY
T1114 LES-CHATS-SAUVAGES 14/04/25 19:30 2 places
T1239 LES-12-HEURES      17/1    12:30 4 places
T32  CAROTTES            22/4/25 18:30 11 places
```

Une commande est constituée exactement ainsi.

1. Sur la première ligne, le mot clé `Dossier` écrit tel quel (minuscules ou majuscules sont importantes) puis le `codeDossier`, un entier composé d'exactly 8 chiffres.
2. Sur la deuxième ligne, nous trouvons le `prenomNom` composé du prénom et nom (écrits en majuscules, sans caractères accentués) séparés par un `/`. Si il y a un nom composé, on utilisera un tiret : un nom ou prénom ne peut commencer ni terminer par un tiret, et on ne doit pouvoir utiliser deux tirets à la suite. On ne met pas de blancs à l'intérieur de `prenomNom`.
3. Puis nous avons une suite de lignes (chacune séparée de la suivante par un seul retour chariot). Nous trouvons sur chaque ligne les informations d'un concert dans cet ordre.
 - (a) le `codeConcert`, code du concert composé de la lettre `T` suivi de 2 à 6 chiffres.
 - (b) le `nomConcert`, nom du concert écrit en majuscules ou chiffres, sans blanc ni ponctuation ni lettre accentuée. Comme pour les noms, on utilisera les tirets pour séparer les mots, sans pouvoir en mettre deux à la suite, et on ne peut mettre de blancs dans le nom du concert.

- (c) une **date** écrite sous le format DD/MM ou DD/M ou DD/MM/YY ou DD/M/YY
- (d) une **heure** écrite sous le format HH:MM
- (e) l'entier **nb** indiquant le nombre de places, cet entier est compris entre 1 et 99,
- (f) le mot clé **places** écrit en minuscules.

4. Enfin la fin de fichier.

L'utilisation du retour chariot est stricte, on ne peut insérer deux retours chariots à la suite, ni grouper deux concerts sur la même ligne. Par ailleurs, ces différents mots sont séparés par des blancs ou des tabulations qui ne jouent que le rôle de séparateur (on peut en mettre plusieurs à la suite sans importance).

Exercice 1 Analyse du lexique d'une commande de billets

1. Déposer sur Updago une spécification **Exo.1.1.1.lex** reconnaissant chacun des éléments (y compris le retour chariot) intervenant dans une commande de billet. Cette spécification devra permettre d'afficher en sortie chaque token reconnu, en éliminant blancs et tabulations en trop. sur l'exemple, la sortie doit être exactement la suivante.

```
dossier codeDossier RC
prenomNom RC
codeConcert nomConcert date heure nb places RC
codeConcert nomConcert date heure nb places RC
codeConcert nomConcert date heure nb places RC
FinFichier
```

2. Compléter la spécification précédente et déposer la spécification **Exo.1.2.1.lex**. Cette spécification permet maintenant de compter le nombre de concerts de chaque commande (l'affichage précédent est supprimé). Lors du traitement du fichier, on doit afficher le *prenomNom* de la personne et le nombre de concerts pour lesquels elle a commandé des places. Sur l'exemple, nous vous afficherez : **ANNE-CLAIRE/BERGEY a acheté des places de 3 concerts.**

Exercice 2 Analyse complète d'une commande de billet

1. Simple contrôle de validité : écrire deux spécifications lex et yacc **Exo.2.1.1.lex** et **Exo.2.1.yacc** permettant de reconnaître une commande de billets. Vous afficherez simplement **La commande est valide** si elle est écrite correctement, le **syntax error** habituel si ce n'est pas le cas suffira.
2. Reprendre et modifier les spécifications précédentes pour écrire deux spécifications lex et yacc **Exo.2.2.1.lex** et **Exo.2.2.yacc** qui cette fois on afficheront le nombre de places concernées par le dossier. Sur l'exemple il faudra afficher : **Le dossier 12345678 concerne 17 places**
3. Pour le moment le format d'une commande ainsi définie ne permet pas d'avoir les numéros de place. On souhaite compléter le format de cette commande pour avoir : pour chaque concert, en dehors du nombre de places, la liste des numéros de places (un numéro de place est un entier non signé).
 - (a) Décrire (en français) comment vous allez rajouter cela à la description d'une commande et modifier votre grammaire, vous mettrez cette explication dans **votreNom.ExplicationPlaces.txt** et écrirez un exemple **votreNom.exempleCommande.txt** de commande de billet avec les numéros des places indiqués.
 - (b) Compléter les spécifications de la question 1 pour obtenir des spécifications **Exo.2.3.1.lex** et **Exo.2.3.yacc** permettant de reconnaître une commande de billets indiquant les numéros de places. (Pas la peine ici d'afficher un résumé de la commande comme dans la question précédente, on se contente juste de savoir si la commande est valide ou si il y a une erreur de syntaxe).