

Examen TLC

NB : Sélectionnez une ou plusieurs réponses (le nombre de réponses n'est pas indiqué).

1) Parmi les propositions suivantes, sélectionnez une ou plusieurs propositions qui correspondent à la démarche correcte à suivre pour écrire un analyseur lexical :

- A - Donner l'automate reconnaissant les unités; puis écrire la fonction de l'analyseur.
- B - Donner les expressions régulières des unités; puis écrire la fonction de l'analyseur.
- C - Ecrire la fonction de l'analyseur; puis donner les expressions régulières des unités.
- D - Ecrire la fonction de l'analyseur; puis donner l'automate reconnaissant les unités.
- E - Donner les expressions régulières des unités et actions en lex; puis produire l'analyseur.

2) Soit le langage L constitué des expressions de la forme

$idf1 = idf2 \text{ op1 } idf3 \text{ op2 } idf4$ ou de la forme $idf1 = idf2 \text{ op1 } idf3$

où les notations $idf1, idf2, idf3, idf4$ dénotent des identificateurs du langage.

Les identificateurs autorisés du langage doivent commencer par une lettre majuscule éventuellement suivie de lettres minuscules.

Les notations $op1$ et $op2$ désignent un opérateur parmi les opérateurs suivants : +, -, *, /

Entourez pour chacune des instructions suivantes le type d'erreur retournée par un compilateur de L :

$Y = a + B$

A- Pas d'erreur B- Erreur lexicale C- Erreur syntaxique

$Y = + A * C$

A- Pas d'erreur B- Erreur lexicale C- Erreur syntaxique

$Y = A / C$

A- Pas d'erreur B- Erreur lexicale C- Erreur syntaxique

$Y = A / C - B$

A- Pas d'erreur B- Erreur lexicale C- Erreur syntaxique

REAL R

BOOLEAN B,C

$R = B * C$

A- Pas d'erreur B- Erreur lexicale C- Erreur syntaxique D- Erreur sémantique

REAL R

BOOLEAN B,C

$R = B ** C$

A- Pas d'erreur B- Erreur lexicale C- Erreur syntaxique D- Erreur sémantique

3) Soit le programme Lex suivant :

```
% {
#include <stdio.h>
% }
%%
"<ul>"      printf("<ol>");
"</ul>"     printf("</ol>");
"bold"      printf("<b>bold</b>");
"italics"   printf("<i>italics</i>");
%%
```

On fournit au programme précédant le texte source suivant en entrée :

```
<html>
<body>
<ul>
<li> le mot bold se trouve dans ce fichier </li>
<li>Le mot <b>bold</b> se trouve dans ce fichier </li>
</ul>
</body>
</html>
```

On souhaite éviter en sortie les balises en double, c-à-d : `body`. Que faut-il ajouter aux règles Lex ?

- A - "`bold`" `printf("bold");`
- B - "``" `printf("");`
- C - "``" `printf("");`
- D - "`bold`" `printf("bold");`

4) Un identificateur est toute chaîne commençant par une lettre suivie éventuellement de lettres ou chiffres. Le programme Lex analyseur.l donné ci-après analyse un texte source pour afficher le nombre d'identificateurs de ce texte, le nombre des entiers, le nombre de caractères et le nombre de lignes comme suit :

nombre de lignes = %d, nombre de cars = %d

```
%option nomain
%option noyywrap
%{
    int num_lignes = 0, num_cars = 0;
%}
%%
\n    ++num_lignes; ++num_cars;
.    ++num_cars;
%%
main() {
    yylex();
    printf( "nombre de lignes = %d, nombre de cars = %d\n", num_lignes,
        num_cars );
}
```

- 5) Le programme précédent est-il complet ?
 - A – Oui
 - B - Non
- 6) Pour compiler le programme précédent, que faut-il faire ?
 - A- lex analyseur
 - B- lex analyseur.l
 - C- lex analyseur.l
gcc analyseur.c -o analyseur
 - D- lex foo.l
 - E- lex foo.l
gcc analyseur
 - F- lex analyseur.l

```
gcc lex.yy.c -o foo
G- lex foo.l
gcc lex.yy.c -o foo
```

- 7) Vous disposez d'un terminal sous Fedora et on souhaite analyser le texte suivant :
- ```
bonjour
lex
au revoir
```

Quel est le résultat qui sera affiché par le programme après l'avoir exécuté ?

- A - nombre d'identificateurs = 4 nombre de caractères = 18 nombre de lignes = 3  
 B - nombre d'identificateurs = 4 nombre de caractères = 21 nombre de lignes = 3  
 C - nombre d'identificateurs = 3 nombre de caractères = 19 nombre de lignes = 3  
 D - nombre d'identificateurs = 4 nombre de caractères = 19 nombre de lignes = 3  
 E - nombre d'identificateurs = 3 nombre de caractères = 22 nombre de lignes = 3  
 F - nombre d'identificateurs = 4 nombre de caractères = 22 nombre de lignes = 3
- 8) L'analyseur lit le texte source depuis l'entrée standard. Que faut-il faire pour pouvoir le lire depuis un fichier appelé texte.txt ?
- A – ajouter yyin = fopen("texte.txt", "r"); avant l'appel à yylex  
 B – ajouter yyout = fopen("texte.txt", "r"); avant l'appel à yylex  
 C – ajouter yyin = fopen("texte.txt", "r"); après l'appel à yylex  
 D – utiliser la redirection '< texte.txt'  
 E – ajouter yyout = fopen("texte.txt", "r"); après l'appel à yylex

- 9) On considère la grammaire G1 des expressions arithmétiques simples

$VT = \{ \text{num } \# \ + \ - \}$

$E \rightarrow \text{num Op F}$

$F \rightarrow E \mid \text{num } \#$

$Op \rightarrow + \mid -$

où num désigne un nombre

Quel est le directeur de  $F \rightarrow E$  ?

A - {num, #, +, - }    B - {num, #, + }    C - {num, # }    D - {num }

- 10) Quel est le directeur de  $F \rightarrow \text{num } \#$  ?

A - {num, #}    B - {num}    C - {#}    D - {+, - }

- 11) La grammaire G1 est-elle LL(1) ?

A – Oui    B – Non

- 12) Soit maintenant la grammaire G2 définie par :

$E \rightarrow \text{num Op num F}$

$F \rightarrow Op E \mid \#$

$Op \rightarrow + \mid -$

- 13) Quel est le nombre des procédures de l'analyseur syntaxique descendant ?

A – 1    B – 2    C – 3    D – 4

- 14) La grammaire est elle LL(1) ?

A – Oui    B – Non