



## Examen de la session de rattrapage (durée : 1h30)

### Questions de cours :

1. Rappelez les phases d'un compilateur et expliquer le rôle de chaque phase en bref.
2. En général un analyseur lexical reconnaît le plus long préfixe qui matche. Comment traiter le cas des commentaires, par exemple en C, de manière à ne pas oublier du code ?

```
void f(int x){ /* une fonction bidon */
    if (x % 2 == 0) x/2 ; else x+3 ;
}/* fin de la fonction bidon */
```

3. Rappelez le principe de la "déterminisation" d'un automate fini non déterministe.

### Exercice 1 :

On considère la grammaire **G** suivante pour des initialisations :

$VAL \rightarrow \{ ' L\_VAL ' \} | ' c '$   
 $L\_VAL \rightarrow LNV | \epsilon$   
 $LNV \rightarrow LNV ' , ' VAL | VAL$

**NB :** VAL désigne la syntaxe que l'on trouve en partie droite de l'affectation d'une initialisation (de tableau ou autre) ; L\_VAL désigne une liste de valeurs (vide ou non) et LNV une liste non vide. 'c' est une unité lexicale correspondant aux constantes.

1. La grammaire G est elle LL(1) ? justifier votre réponse ?
2. Calculer les DEBUTs et SUIVANTs pour les non terminaux de G.
3. Construire la table prédictive de G.
4. Modifier la grammaire G en une grammaire G1 LL(1).
5. Recalculer les DEBUTs et SUIVANTs pour les non terminaux de G1.
6. Construire la table prédictive pour G1.
7. Donner la trace et le résultat de l'analyse de la phrase : { {c1, c2}, {}, c3 }

### Exercice 2 :

Donner le code intermédiaire correspondant au bloc du code en C suivant :

```
{
int i; int j; float[100] a; float v, x;
while(true) {
    do i++; while (a[i]<v);
    do j--; while(a[j]>v);
    if(i>=j) break;
    x=a[i]; a[i]=a[j]; a[j]=x
}
```

### Exercice 3 :

Soit l'expression régulière suivante :

$011(0|1)^*(0|1)^*011$

1. Quel est le langage dénoté par cette expression.
2. Donner l'AFN correspondant à cette expression.
3. Transformer l'AFN de la question 1 en un AFD.
4. Minimiser l'AFD obtenu dans la question 3.