

```

1 Q1.
2
3 S[ts] -> id '++'
4 {
5     place = NouvVar();
6     S.code = (+,place,id,1) @@ (=,ts.place(is.vallex),place, _ )
7 }
8
9 Q2.
10
11 S[ts] -> id '+= ' E
12 {
13     place = NouvVar();
14     S.code = E.code @@ (+,place,id,E.place) @@ (=,ts.place(is.vallex),place, _ )
15 }
16
17 Q3.
18
19 OP : '+' | '*' | '/' | '-' ;
20
21 S[ts] -> id OP '=' E
22 {
23     place = NouvVar();
24     S.code = E.code @@ (OP.vallex,place,id,E.place) @@
25         (=,ts.place(id.vallex),place, _ )
26 }
27
28 Q4.
29
30 S[ts] -> for(S1 ; B ; S2) do LS done
31 {
32     label_debut = nouvLabel();
33     label_fin = nouvLabel();
34     B.siFaux = label_fin;
35     S.code = S1.code @@ (label label_debut, _ , _ , _ ) @@ B.code @@
36         LS.code @@ S2.code @@ (goto label_debut, _ , _ , _ ) @@
37         (label label_fin , _ , _ , _ )
38 }
39
40 Q5.
41 /**
42  * Analyseur syntaxique
43  */
44 object :
45     {' bloc liste_bloc '}'
46 ;
47
48 liste_bloc :
49     ',' bloc liste_bloc
50     | //Mot vide
51 ;
52
53 bloc:
54     object
55     | INT
56     | FLOAT
57     | STRING
58     | BOOLEAN
59     | '[' tab_json ']'
60     | NULL
61 ;
62
63 tab_json :
64     object liste_object
65 ;
66 liste_object :
67     ',' object liste_object
68     | //Mot vide
69 ;
70
71 /**
72  * Analyseur lexical
73  */
74 INT : (0..9)+;
75 FLOAT : (0..9)+ '.' (0..9)+;
76 NULL : 'null';
77 BOOLEAN : 'true' | 'false';
78 STRING : "\"" ('a'..'z' | 'A'..'Z')* "\"" ;

```