**Question B2**

Below is a context-free grammar for a language of assignments that includes arrays:

| | | |
|---|---|---|
| 1. | stmtList | □ stmt stmtList |
| 2. | | □ ε |
| 3. | stmt | □ ID = exp ; |
| 4. | array | □ [ rowList ] |
| 5. | rowList | □ nonEmpty |
| 6. | | □ ε |
| 7. | nonEmpty | □ row moreRows |
| 8. | moreRows | □ ; nonEmpty |
| 9. | | □ ε |
| 10. | row | □ exp more |
| 11. | more | □ , row |
| 12. | | □ ε |
| 13. | exp | □ term tail |
| 14. | tail | □ + term tail |
| 15. | | □ ε |
| 16. | term | □ ID |
| 17. | | □ INTLIT |
| 18. | | □ array |

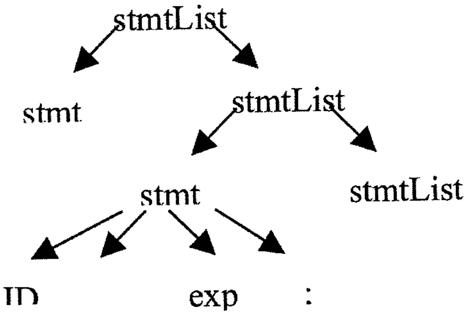
Here are the *FIRST* and *FOLLOW* sets for all of the non-terminals:

| **Non-terminal *X*** | ***FIRST*(*X*)** | ***FOLLOW*(*X*)** |
|---|---|---|
| stmtList | ID ε | EOF |
| stmt | ID | ID EOF |
| array | [ | + , ; ] |
| rowList | ID INTLIT [ ε | ] |
| nonEmpty | ID INTLIT [ | ] |
| moreRows | ; ε | ] |
| row | ID INTLIT [ | ; ] |
| more | , ε | ; ] |
| exp | ID INTLIT [ | , ; ] |
| tail | + ε | , ; ] |
| term | ID INTLIT [ | + , ; ] |

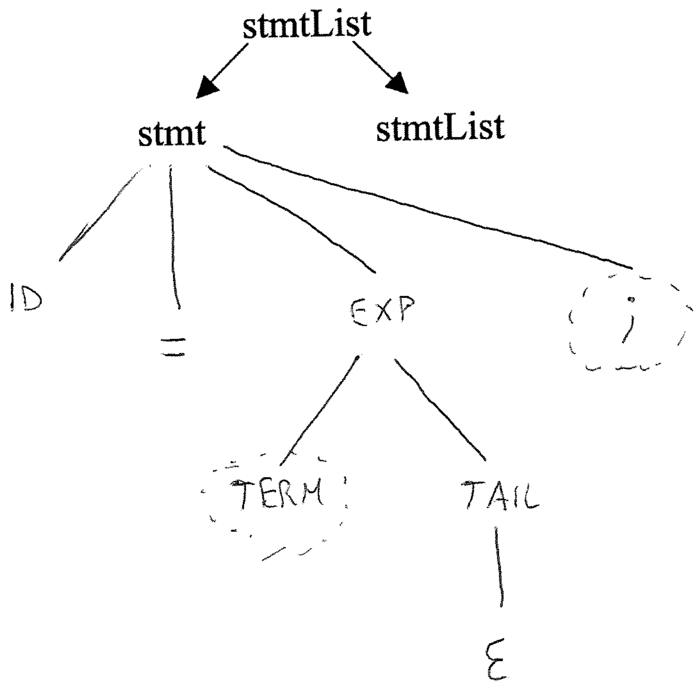
(5 points)

**Part B2a**

Recall that terminal *t* is in *FOLLOW*(*X*) if in some partial parse tree with the start non-terminal at the root, *X* is one leaf of the tree and *t* is the next non-epsilon leaf immediately to the right. For example, the following partial parse tree justifies the fact that for the CFG given above, terminal ID is in *FOLLOW*(stmt):



Complete the partial parse tree below to justify the fact that terminal ; is in *FOLLOW*(term).



(20 points)

**Part B2b**

Fill in the parse table below using the numbers of the grammar rules rather than the rules themselves. Is the grammar LL(1)?

| | ID | INTLIT | = | + | ; | , | [ | ] | EOF |
|---|---|---|---|---|---|---|---|---|---|
| stmtList | 1 | | | | | | | | 2 |
| stmt | 3 | | | | | | | | |
| array | | | | | | | 4 | | |
| rowList | 5 | 5 | | | | | 5 | 6 | |
| nonEmpty | 7 | 7 | | | | | 7 | | |
| moreRows | | | | | 8 | | | 9 | |
| row | 10 | 10 | | | | | 10 | | |
| more | | | | | 12 | 11 | | 12 | |
| exp | 13 | 13 | | | | | 13 | | |
| tail | | | | 14 | 15 | 15 | | 15 | |
| term | 16 | 17 | | | | | 18 | | |

THE GRAMMAR IS LL(1) AS EACH CELL CONTAINS AT MOST 1 PRODUCTION

**Question B3**

Consider the following grammar

$$\begin{array}{rcl}
\mathit{File} & \rightarrow & \mathit{Record} \\
 & | & \mathit{Record}\ \mathit{File} \\
\mathit{Record} & \rightarrow & \textbf{name idnum}\ \mathit{OptGrades} \\
\mathit{OptGrades} & \rightarrow & \mathit{Grades} \\
 & | & \varepsilon \\
\mathit{Grades} & \rightarrow & \mathit{OneGrade} \\
 & | & \mathit{OneGrade}\ \textbf{comma}\ \mathit{Grades} \\
\mathit{OneGrade} & \rightarrow & \textbf{intlit}\ \mathit{OptLate} \\
\mathit{OptLate} & \rightarrow & \mathit{Stars} \\
 & | & \varepsilon \\
\mathit{Stars} & \rightarrow & \textbf{star} \\
 & | & \mathit{Stars}\ \textbf{star}
\end{array}$$

where *File* is the start non-terminal, and symbols in **bold** are terminals.

(10 points)
**Part B3a**
Apply the transformations learned in class to *left factor* the grammar above and write the results below. Give the entire grammar, not the just the transformed rules.

FILE → RECORD FILE1

FILE1 → ε | FILE

RECORD → name idnum OPTGRADES

OPTGRADES → GRADES | ε

GRADES → ONEGRADE G1

G1 → ε | COMMA GRADES

ONEGRADE → INTLIT OPTLATE

OPTLATE → STARS | ε

STARS → STAR | STARS STAR

ID:____________________

(10 points)

**Part B3b**

If the grammar you wrote above has any immediate left recursion, apply the transformation learned in class to remove it and write the result below. You do not need to give the entire grammar, you can just give the transformed rules.

STARS → STAR S'

S' → STAR S' | ε