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:

<table>
<thead>
<tr>
<th>Non-terminal X</th>
<th>FIRST(X)</th>
<th>FOLLOW(X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmtList</td>
<td>ID ε</td>
<td>EOF</td>
</tr>
<tr>
<td>stmt</td>
<td>ID</td>
<td>ID EOF</td>
</tr>
<tr>
<td>array</td>
<td>[</td>
<td>+ ; ]</td>
</tr>
<tr>
<td>rowList</td>
<td>ID INTLIT [ ε ]</td>
<td></td>
</tr>
<tr>
<td>nonEmpty</td>
<td>ID INTLIT [ ]</td>
<td></td>
</tr>
<tr>
<td>moreRows</td>
<td>; ε</td>
<td></td>
</tr>
<tr>
<td>row</td>
<td>ID INTLIT [ ; ]</td>
<td></td>
</tr>
<tr>
<td>more</td>
<td>, ε</td>
<td>;</td>
</tr>
<tr>
<td>exp</td>
<td>ID INTLIT [ ; ]</td>
<td></td>
</tr>
<tr>
<td>tail</td>
<td>+ ε</td>
<td>;</td>
</tr>
<tr>
<td>term</td>
<td>ID INTLIT [ + ; ]</td>
<td></td>
</tr>
</tbody>
</table>

(5 points)
Part B2a
Recall that terminal \( r \) 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 \( FOLLOW(\text{stmt}) \):

```
stmtList
  stmt      stmtList
     stmt
ID      exp
```

Complete the partial parse tree below to justify the fact that terminal ; is in \( FOLLOW(\text{term}) \):

```
stmtList
  stmt      stmtList
     stmt
ID = exp
TERM
\epsilon
```

\( \epsilon \)
(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)?

<table>
<thead>
<tr>
<th></th>
<th>ID</th>
<th>INTLIT</th>
<th>=</th>
<th>+</th>
<th>;</th>
<th>,</th>
<th>[</th>
<th>]</th>
<th>EOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmtList</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>stmt</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>array</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>rowList</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>nonEmpty</td>
<td>7</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>moreRows</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>row</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>more</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>exp</td>
<td>13</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>tail</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>term</td>
<td>16</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
</tbody>
</table>

The grammar is **LL(1)** as each cell contains at most 1 production.
Question B3

Consider the following grammar

\[
\begin{align*}
File & \rightarrow \text{Record} \\
 & \mid \text{Record File} \\
Record & \rightarrow \text{name idnum OptGrades} \\
OptGrades & \rightarrow \text{Grades} \\
 & \mid \epsilon \\
Grades & \rightarrow \text{OneGrade} \\
 & \mid \text{OneGrade comma Grades} \\
OneGrade & \rightarrow \text{intlit OptLate} \\
OptLate & \rightarrow \text{Stars} \\
 & \mid \epsilon \\
Stars & \rightarrow \text{star} \\
 & \mid \text{Stars star}
\end{align*}
\]

where \text{File} is the start non-terminal, and symbols in \textbf{bold} are terminals.

(10 points)
Part B3a

Apply the transformations learned in class to \textit{left factor} the grammar above and write the results below.
Give the entire grammar, not the just the transformed rules.

\[
\begin{align*}
\text{FILE} & \rightarrow \text{RECORD FILE1} \\
\text{FILE1} & \rightarrow \epsilon \mid \text{FILE} \\
\text{RECORD} & \rightarrow \text{name idnum OPTGRADES} \\
\text{OPTGRADES} & \rightarrow \text{GRADES} \mid \epsilon \\
\text{GRADES} & \rightarrow \text{ONEGRADE G1} \\
\text{G1} & \rightarrow \epsilon \mid \text{comma GRADES} \\
\text{ONEGRADE} & \rightarrow \text{intlit OPTLATE} \\
\text{OPTLATE} & \rightarrow \text{STARS} \mid \epsilon \\
\text{STARS} & \rightarrow \text{STAR} \mid \text{STARS STAR}
\end{align*}
\]
(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.

\[
\begin{align*}
S' & \rightarrow \text{star } S' \\
S' & \rightarrow \text{star } S' \mid \epsilon
\end{align*}
\]