1. In general terms, what are the major steps of the typical compiler for an imperative language?  
   Answer: 

2. What is the formal definition of the FIRST(N) for some nonterminal N of a grammar?  
   Answer: 

3. Consider the algorithm to compute CLOSE[I] for the set I of LR(1) items for some grammar. Suppose the grammar contains the production \( X \rightarrow \gamma \) where X is some non-terminal and \( \gamma \) is some string of terminals and non-terminals. Answer the following questions assuming A is some non-terminal, \( \alpha \) and \( \beta \) are strings of terminals and non-terminals, and \( y \) and \( z \) are terminal symbols.  
   (a) If \( A \rightarrow \alpha \bullet X \), \( z \) is in \( I \), which item or items (if any) would be added to CLOSE[I]?  
      Answer: Add \( X \rightarrow \bullet \gamma, z \) to CLOSE[I].  
   (b) If \( A \rightarrow \alpha \bullet X y \), \( z \) is in \( I \), which item or items (if any) would be added to CLOSE[I]?  
      Answer: Add \( X \rightarrow \bullet \gamma, y \) to CLOSE[I].  
   (c) If \( A \rightarrow \alpha \bullet X \beta \), \( z \) is in \( I \), which item or items (if any) would be added to CLOSE[I]?  
      Answer: For any \( w \in \text{FIRST}[\beta z] \), add \( X \rightarrow \bullet \gamma, w \) to CLOSE[I].

4. (Appel, exercise 3.5.) Consider the following grammar. 

```plaintext
1  S  \rightarrow 
2  S  \rightarrow  X S 
3  B  \rightarrow  \{ \text{begin} \{ \text{word} \} \} 
4  E  \rightarrow  \{ \text{end} \{ \text{word} \} \} 
5  X  \rightarrow  B S E 
6  X  \rightarrow  \{ \{ S \} \} 
7  X  \rightarrow  \text{word} 
8  X  \rightarrow  \text{begin} 
9  X  \rightarrow  \text{end} 
10 X  \rightarrow  \{ \text{word} \} 
```
(a) Compute nullable, FIRST, and FOLLOW for all nonterminals of the grammar.

<table>
<thead>
<tr>
<th>nullable</th>
<th>FIRST</th>
<th>FOLLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>yes</td>
<td>$/, {, \text{word, begin, end}$</td>
</tr>
<tr>
<td>$B$</td>
<td>no</td>
<td>$/$</td>
</tr>
<tr>
<td>$E$</td>
<td>no</td>
<td>$a, c$</td>
</tr>
<tr>
<td>$X$</td>
<td>no</td>
<td>$c$</td>
</tr>
</tbody>
</table>

(b) Fill in the LL(1) parse table below for the indicated terminals.

<table>
<thead>
<tr>
<th>word</th>
<th>$/$</th>
<th>${ }$</th>
<th>begin</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B$</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E$</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X$</td>
<td>3 &amp; 10</td>
<td>6</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Answer:

(c) Is the grammar LL(1)?

Answer: No.

5. (Appel, exercise 3.11.) Construct the LR(0) parsing table for the following grammar. And, is the grammar SLR? Explain.

\[
\begin{align*}
0 & \quad S' \rightarrow B \$
\hline
1 & \quad B \rightarrow \text{id} P \\
2 & \quad B \rightarrow \text{id} (E) \\
3 & \quad P \rightarrow \\
4 & \quad P \rightarrow (E) \\
5 & \quad E \rightarrow B \\
6 & \quad E \rightarrow B, E
\end{align*}
\]
6. (Appel, exercise 3.14.) Construct the LR(1) parsing table for the following grammar.

\[
\begin{align*}
0 & \quad S' \rightarrow S \$ \\
1 & \quad S \rightarrow (X \\
2 & \quad S \rightarrow E] \\
3 & \quad S \rightarrow F) \\
4 & \quad X \rightarrow E) \\
5 & \quad X \rightarrow F] \\
6 & \quad E \rightarrow A \\
7 & \quad F \rightarrow A \\
8 & \quad A \rightarrow \\
\end{align*}
\]