1) 1) Consider the grammar

\[
E \rightarrow E \text{ OR } T \mid T \\
T \rightarrow T \text{ and } F \mid F \\
F \rightarrow \text{not } F \mid (E) \mid \text{true} \mid \text{false}
\]

a) Construct a parse tree for sentence

\[\text{not (true or false)}\]

Answer: You do it

b) Show that this generates all Boolean expressions.

Answer: First show that the grammar generates boolean expressions.
Second show that, given any Boolean expression this grammar can generate it.

FOCUS ON RIGHT MOST DERIVATIONS.

Proof: (First). Induction on the length of the strings derived.

Induction Hypothesis P(n): The grammar generates a string of length \(\leq n\) then its a Boolean expressions.

Let \(n = 1\) (base case)
The only strings on length 1 are true and false. Hence base case is true.

Let \($n > 1$ assume P(n). Now consider a string of length \(n+1\).

Case 1. Either the string ends with a \(\text{OR true}\) or \(\text{OR false}\). If this happens then the derivation was due to the production 
\(E \rightarrow E \text{ OR } T;\) now clearly the string generated by \(E\) is length \((n-1)\), by Induction hypothesis
the string of length(n+1 is a Boolean expression

case 2. The string end with AND true or AND false.
Then the only way this could happen is
E --> T --> T AND F
Again since T derives strings of length (n-1)
and E --> T we are done.


The only way this can happen is
F --> ( E ). clearly, E derives strings
of length n-1, we are done.

Now to show the second part, use induction. That is given a string
which is Boolean expression, show by induction on the length of the
string you can derive it.

c) Is this grammar ambiguous?

NO

d) Construct a LL(1) table for this grammar. Are there any
conflicts in the table?

First(E) = First(T) = First(F) = { true, false, (, not }
Follow(E) = { OR, ),
Follow(T) = { AND, OR, ) } = Follow(F)
Note: We don't need the follows since we don't have
any ε derivations.

Lets number the productions:

1) E --> E OR T
2) E --> T
3) T --> T and F
4) T --> F
5) F --> not F
6) F-->( E )
7) F--> true
8) F--> false

<table>
<thead>
<tr>
<th></th>
<th>OR</th>
<th>AND</th>
<th>TRUE</th>
<th>FALSE</th>
<th>NOT</th>
<th>(</th>
<th>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td>1,2</td>
<td>1,2</td>
<td>1,2</td>
<td>1,2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>3,4</td>
<td>3,4</td>
<td>3,4</td>
<td>3,4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are conflicts in the table.