1. Given the following grammar for identifiers (Id):

\[
\begin{align*}
\text{Id} & \rightarrow \text{Let} | \text{Id Let} | \text{Id Dig} \\
\text{Let} & \rightarrow \text{a } | \text{b } | \text{c} \\
\text{Dig} & \rightarrow \text{0 } | \text{1}
\end{align*}
\]

write a regular expression defining identifiers

2. Given the following two grammars for matching parentheses

Grammar 1: \( S = ( ) | ( S ) \)

Grammar 2: \( M = ( ) | ( M ) | M \ M \)

2a. Show a derivation of \( (( () )) \) using Grammar 1, starting with \( S \)

2b. Show a derivation of \( ( () ) ( \) using Grammar 2, starting with \( M \)
2c. Is ( ) () produced by
1. Grammar 1? (Y/N)
2. Grammar 2? (Y/N)

2d. Is ( ) () produced by
1. Grammar 1? (Y/N)
2. Grammar 2? (Y/N)

3. Complete the following table, keeping the operands in the same order

<table>
<thead>
<tr>
<th>Prefix expression</th>
<th>Infix expression</th>
<th>Postfix expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>* + a b c</td>
<td>(a+b) * c</td>
<td>a b + c *</td>
</tr>
<tr>
<td></td>
<td>a - b + c - d</td>
<td></td>
</tr>
<tr>
<td>* / a b + c d</td>
<td></td>
<td>a b c d - - *</td>
</tr>
<tr>
<td></td>
<td>true or true and false</td>
<td></td>
</tr>
</tbody>
</table>