1) Number representation

a) Represent these numbers in 6-bit signed 2’s complement form or indicate that it is not possible.

<table>
<thead>
<tr>
<th></th>
<th>111111</th>
<th>110001</th>
<th>not possible</th>
<th>not possible</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>11111</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-7</td>
<td>11100</td>
<td>11000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>not possible</td>
<td>10111</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>11011</td>
<td></td>
<td></td>
<td></td>
<td>10000</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td>01110</td>
<td></td>
<td></td>
<td>100000</td>
</tr>
</tbody>
</table>

b) For each binary vector below, what does it represent in Octal and Hexadecimal?

<table>
<thead>
<tr>
<th>Binary</th>
<th>Octal</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>101111</td>
<td>57</td>
<td>2F</td>
</tr>
<tr>
<td>110000</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>011110</td>
<td>36</td>
<td>1E</td>
</tr>
</tbody>
</table>
2) Two-level logic minimization

\[ F(A, B, C, D) = \sum m(1, 6, 7, 10, 11, 12, 13) \]
\[ d(A, B, C, D) = \sum m(0, 3, 5, 9, 15) \]

a) Identify all the prime and essential prime implicants.

<table>
<thead>
<tr>
<th>Primes (including Essentials)</th>
<th>Essential Primes Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bar{D} )</td>
<td>( X )</td>
</tr>
<tr>
<td>( \bar{A} \bar{B} \bar{C} )</td>
<td>( X )</td>
</tr>
<tr>
<td>( \bar{A} \bar{B} \bar{C} )</td>
<td>( X )</td>
</tr>
<tr>
<td>( \bar{A} \bar{B} \bar{C} )</td>
<td>( X )</td>
</tr>
</tbody>
</table>

b) Find the minimum two-level logic implementation.

\[ F = \bar{A} \bar{B} \bar{C} + A \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C} + D \]
3) Multi-level logic minimization

Given following logic equations, minimize the number of literals, e.g. by using common subexpressions, Boolean rules, etc. You can introduce new immediate equations for common sub-expressions if it helps to reduce the number of literals. Put a box around your final answer, and indicate the number of literals in your final answer.

\[
P = (\bar{A} + \bar{B} + \bar{C})(D + C + \bar{C}) \\
   = (A + B + C)(D + 1) \\
   = \bar{A} + \bar{B} + \bar{C}
\]

\[
Q = ABC\bar{D} + ABC\bar{E} + ABC\bar{F} \\
   = ABC(\bar{D} + \bar{E} + \bar{F}) \\
   = \bar{P}\bar{R}
\]

\[
R = \bar{B}DEF + CDEF + DEF \\
   = DEF(\bar{B} + C + 1) \\
   = DEF
\]

\[
S = A\bar{B}(\bar{A} + B) \\
   = 0
\]

\[
T = DEF + (\bar{A} + \bar{B} + \bar{C})(\bar{D} + \bar{E} + \bar{F}) \\
   = R + P\bar{R} \\
   = R + P \\
   = \bar{Q}
\]

9 literals in total.
4) Delay analysis

Consider the following 8-bit Carry Select Adder design that adds the numbers $F = X + Y$. Assume 1\text{ns} gate delay for all gates (i.e., 2-XOR, 2-AND, 2-OR, and 2-input MUX).

What are final delays for the following:

- $S_0 = 2\text{ns}$
- $S_2 = 6\text{ns}$
- $S_4 = 10\text{ns}$
- $S_6 = 10\text{ns}$
- $S_8 = 10\text{ns}$
- $S_1 = 4\text{ns}$
- $S_3 = 8\text{ns}$
- $S_5 = 10\text{ns}$
- $S_7 = 10\text{ns}$

The logic for each bitslice (rectangle box) is as follows: