Note that in many questions there are multiple different solutions.

**Question 1:**
(a) 1100001
(b) 167_{10}
(c) 48_{10}
(d) 0x3B26

**Question 2:**
(a) 

(b) 

(c) \neg (A \land C)

**Question 3:**
(a) 

(b) 

OR
\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 1 & 0 & 1 \\
\times & 0 & 0 & 0 & 1 & 0 & 1 \\
\hline
1 & 0 & 1 & 1 & \\
0 & 0 & 0 & 0 & 0 & \\
\hline
+ & 1 & 0 & 1 & 1 & 0 & 0 \\
\hline
0 & 0 & 1 & 1 & 0 & 1 & 1 \\
\end{array}
\]

(c)

Question 4: (a) \( NaN \) (b) \(-6.5_{10}\) (c) \( 0101000111 \) (d) \( 0000000101 \)

Question 5:

\begin{verbatim}
addi $3, $0, 10
div $1, $3
mflo $4
mfhi $2

div $4, $3
mflo $4
mfhi $5
add $2, $2, $5
add $2, $2, $4

jr $31
\end{verbatim}

Question 6:

\begin{verbatim}
vehicleweight:
addi $8, $0, 6
slt $9, $4, $8
beq $9, $0, truck

addi $30, $30, $-4
sw $31, 0($30)
jal carweight
lw $31, 0($30)
addi $30, $30, 4

beq $0, $0, end
\end{verbatim}

(continues on next page)
truck:
    addi $30, $30, -4
    sw $31, 0($30)
    jal truckweight
    lw $31, 0($30)
    addi $30, $30, 4
end:
    jr $31

Question 7:

<table>
<thead>
<tr>
<th>Value</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>-12</td>
<td>0x0000008C</td>
</tr>
<tr>
<td>-14</td>
<td>0x00000090</td>
</tr>
<tr>
<td>-7</td>
<td>0x00000094</td>
</tr>
<tr>
<td>2</td>
<td>0x00000098</td>
</tr>
<tr>
<td>-1</td>
<td>0x000000A0</td>
</tr>
<tr>
<td>4</td>
<td>0x000000A4</td>
</tr>
</tbody>
</table>

Question 8:

(a) Underflow occurs when a number is too close to zero to be represented by a normal floating point number. This is problematic because the number becomes zero rather than a small fraction, which may then go on to affect later calculations. Subnormal numbers help solve this problem by adding additional numbers closer to zero. It does not fully solve the problem, as smaller numbers will always still exist. Adjusting the bias may also help mitigate this problem, at the expense of larger numbers.

(b) In one’s complement, to negate a number you flip the bits. In two’s complement you flip the bits and add one.

Advantage 1: Two’s complement only has one zero.
Advantage 2: $n$-bit two’s complement has the range $(-2^{n-1}, 2^{n-1} - 1)$, but $n$-bit one’s complement only has the range $(-2^{n-1} + 1, 2^{n-1} - 1)$.

Advantage 3: Binary arithmetic works unchanged on two’s complement numbers.
(c) In a normal ripple carry adder, the most significant bit has to wait for the carry to propagate all the way from the least significant bit before it can actually finish the addition. With a carry lookahead, the carries for more significant bits can be computed directly and the addition can proceed in parallel.