Please write neatly and clearly. Make sure you have 4 questions over 6 pages (the last page lists some IA32 instructions). You have 75 minutes to answer all questions.
1) Assume the variables \( a \) and \( b \) are signed integers and that the machine uses two’s complement representation. Also assume that \( \text{MAX\_INT} \) is the maximum integer, \( \text{MIN\_INT} \) is the minimum integer, and \( W \) is one less than the word length (e.g., \( W = 31 \) for 32-bit integers).

Match each of the descriptions on the left with a line of code on the right – write in (i), (ii), etc.

Note that \( (a < 0) \ ? \ 1 : -1 \) evaluates to 1 if \( a < 0 \), otherwise it evaluates to -1.

<table>
<thead>
<tr>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( \neg a ).</td>
<td>(i) ( \neg^2 a \ &amp; \ (\neg b \ ^{\neg} (\text{MIN_INT} + \text{MAX_INT})) )</td>
</tr>
<tr>
<td>2. ( b ).</td>
<td>(ii) ( ((a ^{\neg} b) \ &amp; \ ^{\neg} a) \</td>
</tr>
<tr>
<td>3. ( a \</td>
<td>\ ^{\neg} b ).</td>
</tr>
<tr>
<td>4. ( a \ * \ 7 ).</td>
<td>(iv) (a \text{ &lt;&lt; 4}) + (a \text{ &lt;&lt; 2}) + (a \text{ &lt;&lt; 1})</td>
</tr>
<tr>
<td>5. ( (a &lt; 0) \ ? \ 1 : -1 ).</td>
<td>(v) ( ((a &lt; 0) \ ? \ (a + 3) : a) \text{ &gt;&gt; 2} )</td>
</tr>
<tr>
<td></td>
<td>(vi) ( a \ ^{\neg} (\text{MIN_INT} + \text{MAX_INT}) )</td>
</tr>
<tr>
<td></td>
<td>(vii) ( \neg((a \</td>
</tr>
<tr>
<td></td>
<td>(viii) ( \neg((a \text{ &gt;&gt; W}) \text{ &lt;&lt; 1}) )</td>
</tr>
</tbody>
</table>
NAME:

2) Match each of the assembler routines on the left with the equivalent C function on the right. Recall an address operand of the form \( D(R_b,R_i,s) \) evaluates to \( (R_b)+(R_i)\times s+D \), where \( R_b \) and \( R_i \) represent registers, \( D \) and \( s \) represent constants, and \( R \) refers to the contents of register \( R \). Also recall that \texttt{sall} denotes a shift arithmetic left instruction, \texttt{shrl} denotes a shift logical right, and \texttt{leal} denotes a load effective address instruction (see last page).

foo1:
\[
\begin{align*}
\text{pushl} & \ %ebp \\
\text{movl} & \ %esp,%ebp \ // \text{set up} \\
\text{movl} & \ 8(%ebp),%eax \\
\text{sall} & \ $4,%eax \\
\text{subl} & \ 8(%ebp),%eax \\
\text{movl} & \ %ebp,%esp \ // \text{finish} \\
\text{popl} & \ %ebp \\
\text{ret}
\end{align*}
\]

foo2:
\[
\begin{align*}
\text{pushl} & \ %ebp \\
\text{movl} & \ %esp,%ebp \ // \text{set up} \\
\text{movl} & \ 8(%ebp),%eax \\
\text{leal} & \ $4(%eax,%eax,2),%eax \\
\text{movl} & \ %ebp,%esp \ // \text{finish} \\
\text{popl} & \ %ebp \\
\text{ret}
\end{align*}
\]

foo3:
\[
\begin{align*}
\text{pushl} & \ %ebp \\
\text{movl} & \ %esp,%ebp \ // \text{set up} \\
\text{movl} & \ 8(%ebp),%eax \\
\text{shrl} & \ $31,%eax \\
\text{movl} & \ %ebp,%esp \ // \text{finish} \\
\text{popl} & \ %ebp \\
\text{ret}
\end{align*}
\]

int choice1(int x)
{ 
    return (x < 0);
}

int choice2(int x)
{ 
    return (x << 31) & 1;
}

int choice3(int x)
{ 
    return 15 * x;
}

int choice4(int x)
{ 
    return (x + 15) /4
}

int choice5(int x)
{ 
    return 3*x + 4;
}

int choice6(int x)
{ 
    return (x >> 31);
}

\textbf{Fill in your answers here:}
foo1 corresponds to choice ________.
foo2 corresponds to choice ________.
foo3 corresponds to choice ________.
3) Consider a 5-bit two’s complement representation. Denote by $T_{\text{Max}}$ the maximum representable number (integer), and by $T_{\text{Min}}$ the minimum number. Fill in the empty boxes in the following table. Addition and subtraction should be performed based on the rules for 5-bit, two’s complement arithmetic.

<table>
<thead>
<tr>
<th>Number</th>
<th>Decimal Representation</th>
<th>Binary Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>n/a</td>
<td>$-2$</td>
<td></td>
</tr>
<tr>
<td>n/a</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>n/a</td>
<td></td>
<td>0 1100</td>
</tr>
<tr>
<td>n/a</td>
<td></td>
<td>1 0100</td>
</tr>
<tr>
<td>$T_{\text{Max}}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{\text{Min}}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{\text{Min}} + T_{\text{Min}}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{\text{Min}} + 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{\text{Max}} + 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(a) What are the largest and smallest positive (normalized) values that can be represented?

(b) What does the expression \( x + (y + z) \) evaluate to, if \( x = 3.5_{10} \), \( y = -3.5_{10} \), and \( z = 0.25_{10} \)? Justify your answer by showing the internal bit representations during the evaluation steps of this expression. Assume that during the evaluation steps, extra precision bits that do not fit within the given precision of 3-bit fraction are truncated (dropped).
Cheat Sheet

We have covered the following Intel IA32 instructions in class so far.

```
movl Src,Dest  # If Src = address p, Dest = *p
leal Src,Dest  # Dest = Src, where Src is address expression

addl Src,Dest  # Dest = Dest + Src
subl Src,Dest  # Dest = Dest - Src
imull Src,Dest  # Dest = Dest * Src
sall Src,Dest  # Dest = Dest << Src Also called shll
sar1 Src,Dest  # Dest = Dest >> Src Arithmetic
shrl Src,Dest  # Dest = Dest >> Src Logical
xorl Src,Dest  # Dest = Dest ^ Src
andl Src,Dest  # Dest = Dest & Src
orl Src,Dest  # Dest = Dest | Src

incl Dest     # Dest = Dest + 1
dectl Dest     # Dest = Dest - 1
negl Dest      # Dest = - Dest
notl Dest      # Dest = ~ Dest
```