they are quite convenient and result in more readable code (for example, the \( i \) and move instructions). If you choose to use pseudoinstructions for these reasons, please add a sentence or two to your solution stating which pseudoinstructions you have used and why.

**Exercise 2.1**

The following problems explore translating from C to MIPS. Assume that the variables \( f, g, h, \) and \( i \) are given and could be considered 32-bit integers as declared in a C program.

\[
\begin{align*}
\text{a.} & \quad f = g - h; \\
\text{b.} & \quad f = f + (h - 5); \\
\end{align*}
\]

2.1.1 [5] <2.2> For the C statements above, what is the corresponding MIPS assembly code? Use a minimal number of MIPS assembly instructions.

2.1.2 [5] <2.2> For the C statements above, how many MIPS assembly instructions are needed to perform the C statement?

2.1.3 [5] <2.2> If the variables \( f, g, h, \) and \( i \) have values 1, 2, 3, and 4, respectively, what is the end value of \( f \)?

The following problems deal with translating from MIPS to C. Assume that the variables \( g, h, i, \) and \( j \) are given and could be considered 32-bit integers as declared in a C program.

\[
\begin{align*}
\text{a.} & \quad \text{add} f, f, 4 \\
\text{b.} & \quad \text{add} f, g, h \\
& \quad \text{add} f, i, f \\
\end{align*}
\]

2.1.4 [5] <2.2> For the MIPS assembly instructions above, what is a corresponding C statement?

2.1.5 [5] <2.2> If the variables \( f, g, h, \) and \( i \) have values 1, 2, 3, and 4, respectively, what is the end value of \( f \)?

**Exercise 2.2**

The following problems deal with translating from C to MIPS. Assume that the variables \( g, h, i, \) and \( j \) are given and could be considered 32-bit integers as declared in a C program.
2.8.4 [5] <2.4> Assume that register $s0 = 0x70000000$ and $s1 = 0x10000000$. For the table above, will there be overflow?

2.8.5 [5] <2.4> Assume that register $s0 = 0x40000000$ and $s1 = 0x20000000$. For the table above, will there be overflow?

2.8.6 [5] <2.4> Assume that register $s0 = 0x8FFFFFFF$ and $s1 = 0xD0000000$. For the table above, will there be overflow?

Exercise 2.9

The table below contains various values for register $s1$. You will be asked to evaluate if there would be overflow for a given operation.

| a. | (1) $s0$ |
| b. | 1024, $s0$ |

2.9.1 [5] <2.4> Assume that register $s0 = 0x70000000$ and $s1$ has the value as given in the table. If the instruction: add $s0$, $s0$, $s1$ is executed, will there be overflow?

2.9.2 [5] <2.4> Assume that register $s0 = 0x80000000$ and $s1$ has the value as given in the table. If the instruction: sub $s0$, $s0$, $s1$ is executed, will there be overflow?

2.9.3 [5] <2.4> Assume that register $s0 = 0x7FFFFFFF$ and $s1$ has the value as given in the table. If the instruction: sub $s0$, $s0$, $s1$ is executed, will there be overflow?

The table below contains various values for register $s1$. You will be asked to evaluate if there would be overflow for a given operation.

| a. | 0010 0100 1001 0010 0100 1001 0010 0100 $s0$ |
| b. | 0111 1111 1011 1110 0100 0000 0000 0000 $s0$ |

2.9.4 [5] <2.4> Assume that register $s0 = 0x70000000$ and $s1$ has the value as given in the table. If the instruction: add $s0$, $s0$, $s1$ is executed, will there be overflow?
2.9.5 [5] <2.4> Assume that register $s0 = 0x70000000$ and $s1$ has the value as given in the table. If the instruction: add $s0$, $s0$, $s1$ is executed, what is the result in hex?

2.9.6 [5] <2.4> Assume that register $s0 = 0x70000000$ and $s1$ has the value as given in the table. If the instruction: add $s0$, $s0$, $s1$ is executed, what is the result in base ten?

**Exercise 2.10**

In the following problems, the data table contains bits that represent the opcode of an instruction. You will be asked to interpret the bits as MIPS instructions into assembly code and determine what format of MIPS instruction the bits represent.

| a. | 0000 0010 0001 0000 1000 0000 0010 0000 |
| b. | 0000 0001 0100 1011 0100 1000 0010 0010 |

2.10.1 [5] <2.5> For the binary entries above, what instruction do they represent?

2.10.2 [5] <2.5> What type (I-type, R-type, J-type) instruction do the binary entries above represent?

2.10.3 [5] <2.4, 2.5> If the binary entries above were data bits, what number would they represent in hexadecimal?

In the following problems, the data table contains MIPS instructions. You will be asked to translate the entries into the bits of the opcode and determine the MIPS instruction format.

| a. | addi $t0$, $t0$, 0 |
| b. | sw $t1$, 32($t2$) |

2.10.4 [5] <2.4, 2.5> For the instructions above, show the binary then hexadecimal representation of these instructions.

2.10.5 [5] <2.5> What type (I-type, R-type, J-type) instruction do the instructions above represent?

2.10.6 [5] <2.5> What is the binary then hexadecimal representation of the opcode, Rs, and Rt fields in this instruction? For R-type instructions, what is the hexadecimal representation of the rd and funct fields? For I-type instructions, what is the hexadecimal representation of the immediate field?
2.13.3 [5] <2.6> For the lines above, what is the value of $t2$ for the following sequence of instructions?

```
srl $t2, $t0, 3
andi $t2, $t2, 0xFFFF
```

In the following exercise, the data table contains various MIPS logical operations. You will be asked to find the result of these operations given values for registers $t0$ and $t1$.

|   | $s1$ | $s2$, $t0$, 1
|---|------|------------------
| a | srl  | $t2$, $t2$, -1
| b | andi | $t2$, $t1$, 0x0000
|   | srl  | $t2$, 2

2.13.4 [5] <2.6> Assume that $t0 = 0x0000A5A5$ and $t1 = 00005A5A$. What is the value of $t2$ after the two instructions in the table?

2.13.5 [5] <2.6> Assume that $t0 = 0xA5A50000$ and $t1 = A5A50000$. What is the value of $t2$ after the two instructions in the table?

2.13.6 [5] <2.6> Assume that $t0 = 0xA5A5FFFF$ and $t1 = A5A5FFFF$. What is the value of $t2$ after the two instructions in the table?

**Exercise 2.14**

The following figure shows the placement of a bit field in register $t0$.

```
  31  30  29  28  27  26  25  24  23  22  21  20  19  18  17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0

Field

^31 - i bits

    i - j bits

      j bits
```

In the following problems, you will be asked to write MIPS instructions to extract the bits “Field” from register $t0$ and place them into register $t1$ at the location indicated in the following table.

<table>
<thead>
<tr>
<th></th>
<th>31</th>
<th>31 - (i - j)</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>31</td>
<td>Field</td>
<td>0 0 0...0 0 0</td>
</tr>
<tr>
<td>b</td>
<td>31</td>
<td>14 + i - j bits</td>
<td>14</td>
</tr>
<tr>
<td>111 ... 111</td>
<td>Field</td>
<td>111 ... 111</td>
<td></td>
</tr>
</tbody>
</table>
2.14.1 [20] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from $t0 for the constant values $i = 22$ and $j = 5$ and places the field into $tl$ in the format shown in the data table.

2.14.2 [5] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from $t0 for the constant values $i = 4$ and $j = 0$ and places the field into $tl$ in the format shown in the data table.

2.14.3 [5] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from $t0 for the constant values $i = 31$ and $j = 28$ and places the field into $tl$ in the format shown in the data table.

In the following problems, you will be asked to write MIPS instructions to extract the bits “Field” from register $t0 shown in the figure and place them into register $tl$ at the location indicated in the following table. The bits shown as “XXX” are to remain unchanged.

<table>
<thead>
<tr>
<th></th>
<th>31</th>
<th>31 - (i - j)</th>
<th>XXX ... XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td></td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td>14 + i - j bits</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>XXX ... XXX</td>
<td>Field</td>
<td>XXX ... XXX</td>
</tr>
</tbody>
</table>

2.14.4 [20] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from $t0 for the constant values $i = 17$ and $j = 11$ and places the field into $tl$ in the format shown in the data table.

2.14.5 [5] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from $t0 for the constant values $i = 5$ and $j = 0$ and places the field into $tl$ in the format shown in the data table.

2.14.6 [5] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from $t0 for the constant values $i = 31$ and $j = 29$ and places the field into $tl$ in the format shown in the data table.

Exercise 2.15

For these problems, the table holds some logical operations that are not included in the MIPS instruction set. How can these instructions be implemented?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>not $tl$, $t2$ // bit-wise invert</td>
</tr>
<tr>
<td>b.</td>
<td>orn $tl$, $t2$, $t3$ // bit-wise OR of $t2$, $t3$</td>
</tr>
</tbody>
</table>

2.15 | instr value
2.15 | instructions instructions instr
2.15 | repre Vario will $tl$ MIPS

2.15 | oper and $tl$ repre
2.15 | seque $tl$ repre

2.15 | Exe For $tl$ the $v$

2.16 | has $tl$