#1. What is the value (in hex) of %o1 after each set of instructions:

a) set 0xDEADBEEF, %o1
sra %o1, 12, %o1

Value in %o1 at this point is 0x______________

b) set 0xDEADBEEF, %o1
set 0x13579246, %o2
xor %o1, %o2, %o1

Value in %o1 at this point is 0x______________

c) set 0xDEADBEEF, %o1
set 0x13579246, %o2
and %o1, %o2, %o1

Value in %o1 at this point is 0x______________

#2. Fill in the blanks to correctly implement the following C code fragment in SPARC assembly.
Assume arg1 and arg2 are formal parameters available in the first two in registers.
Assume x is mapped to local register 0. No optimization of delay slots.

```
if ( arg1 < arg2 )
    x = 37;
else
    x = 42;
```

```
_____ %i0, %i1
_____ L5
_____  
_____ mov 37, %10 ! x = 37;
_____ L3
_____ 
_____ : 
_____ mov 42, %10 ! x = 42;
_____ :
```
#3. Write the equivalent **unoptimized** SPARC assembly language instructions to perform the following C code fragment. **Use the loop construct specified in class/Notes.**

<table>
<thead>
<tr>
<th>C</th>
<th>SPARC assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>for ( a = 432; a &gt;= 77; --a )</td>
<td>/* x is mapped to %l0 */</td>
</tr>
<tr>
<td>{</td>
<td>/* a is mapped to %l7 */</td>
</tr>
<tr>
<td>x = a - 37;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

#4a. Write the equivalent **unoptimized** SPARC assembly language instructions to perform the following C code fragment.

<table>
<thead>
<tr>
<th>C</th>
<th>SPARC assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = x * 12345;</td>
<td>/* x is mapped to %l0 */</td>
</tr>
</tbody>
</table>

4b. Now optimize your answer from #4a to eliminate any delay slots:

**Optimized version of above SPARC assembly**