#1. a) Write the SPARC assembly instructions to define the following *global* variables in the *data* segment:

```assembly
c char ch = 'a';
int i = 405;
char *fmt = "Hello World\n";
```

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

a) 
```
set 0xDEADF00D, %o1
sll %o1, 12, %o1
```
Value in %o1 at this point is \textbf{0x}______________________________

b) 
```
set 0xDEADF00D, %o1
sra %o1, 20, %o1
```
Value in %o1 at this point is \textbf{0x}______________________________

c) 
```
set 0xDEADF00D, %o1
set 0x9C9C9C9C, %o2
and %o1, %o2, %o1
```
Value in %o1 at this point is \textbf{0x}______________________________

d) 
```
set 0xDEADF00D, %o1
set 0x9C9C9C9C, %o2
btog %o2, %o1
```
Value in %o1 at this point is \textbf{0x}______________________________
#3. 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 = -8118;</td>
<td>/* x is mapped to %l4 */</td>
</tr>
<tr>
<td>do {</td>
<td></td>
</tr>
<tr>
<td>x = x + 15;</td>
<td></td>
</tr>
<tr>
<td>--x;</td>
<td></td>
</tr>
<tr>
<td>} while( x &lt;= 420 );</td>
<td></td>
</tr>
</tbody>
</table>

Now optimize your answer to eliminate any delay slots:

**Optimized version of above SPARC assembly**