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

```assembly
char CSI[] = "Miami";
short hair = 911;
```

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

a) set 0x8630131B, %o1
    sra %o1, 24, %o1

Value in %o1 at this point is **0x**

b) set 0x8630131B, %o1
    set 0x5A5A5A5A, %o2
    xor %o1, %o2, %o1

Value in %o1 at this point is **0x**

c) set 0x8630131B, %o1
    set 0x5A5A5A5A, %o2
    and %o1, %o2, %o1

Value in %o1 at this point is **0x**

#3. Assume you run gdb on pa1.
State how to set a breakpoint at the entry point in displayX():

Assume you correctly set this breakpoint and performed a run with correct command line arguments.
State how to print the value of the 1st argument passed to displayX():
#4. Write the equivalent **unoptimized** SPARC assembly language instructions to perform the following C code fragment. **Use the loop construct specified in class/Notes.**

```
for ( x = 14; x >= 129; ++x )
{
    a = x - 8008;
}
```

**SPARC assembly**

/* x is mapped to %l2 */
/* a is mapped to %l4 */

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

```
x = x / 9009;
```

**SPARC assembly**

/* x is mapped to %l0 */

Now optimize your answer to eliminate any delay slots:

**Optimized version of above SPARC assembly**