#1. a) Write the SPARC assembly instructions to define the following global variables in the data segment:
   
   ```
   char May18[] = "Sungod";
   short stack = 17;
   ```

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

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

   Value in %o1 at this point is 0x______________________________

   b)   set 0xCafeBABe, %o1
        set 0x8642ABCD, %o2
        xor %o1, %o2, %o1

   Value in %o1 at this point is 0x______________________________

   c)   set 0xCafeBABe, %o1
        set 0x8642ABCD, %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 2nd argument passed to displayX() in gdb:

(over)
#4. Write the equivalent unoptimized SPARC assembly language instructions to perform the following C code fragment. Use the loop construct specified in class/Notes.

C

```c
for ( x = 421; x > 37; --x )
{
    a = x - 42;
}
```

SPARC assembly

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

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

C

```c
x = x * 7777;
```

SPARC assembly

```assembly
/* x is mapped to %l0 */
```

5b. Now optimize your answer from #5a to eliminate any delay slots:

Optimized version of above SPARC assembly