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

```c
char GreenDay[] = "Jesus of Suburbia";
double earthday = 4.22;
```

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

**a)**

```c
set 0x867F5309, %o1
sra %o1, 12, %o1
```

Value in %o1 at this point is 0x___________________________

**b)**

```c
set 0x867F5309, %o1
set 0xB5B5B5B5, %o2
xor %o1, %o2, %o1
```

Value in %o1 at this point is 0x___________________________

**c)**

```c
set 0x867F5309, %o1
set 0xB5B5B5B5, %o2
and %o1, %o2, %o1
```

Value in %o1 at this point is 0x___________________________

#3. List two reasons why a local variable cannot be mapped to a local register and must be allocated on the runtime stack in the SPARC architecture.

1) 

2) 

(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
for ( i = 5678; i > 317; --i ) {
    x = i + 123;
}

SPARC assembly
/* i is mapped to %l3 */
/* x is mapped to %l5 */

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

C
x = x % 7777;

SPARC assembly
/* x is mapped to %l5 */

Now optimize your answer to eliminate any delay slots:

Optimized version of above SPARC assembly