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

```asm
char Coachella[] = "This Weekend";
double tunes = 94.9;
```

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

a) set 0xFACEBABE, %o1
    sra %o1, 16, %o1

Value in %o1 at this point is 0x__________________________

b) set 0xFACEBABE, %o1
    set 0x4D3C2B1A, %o2
    xor %o1, %o2, %o1

Value in %o1 at this point is 0x__________________________

c) set 0xFACEBABE, %o1
    set 0x4D3C2B1A, %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 = 7154; x < 181; --x )
{
    a = x + 543;
}
```

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 % 7956;
```

SPARC assembly

```assembly
/* x is mapped to %10 */
```

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

**Optimized version of above SPARC assembly**