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

```assembly
char Terminator[] = "Sarah Conner";
double DSC = 101.5;
```

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

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

Value in %o1 at this point is 0x______________________________

b) set 0xDEADBEEF, %o1
    set 0x1A2A3A4A, %o2
    xor %o1, %o2, %o1

Value in %o1 at this point is 0x______________________________

c) set 0xDEADBEEF, %o1
    set 0x1A2A3A4A, %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 displayDiamond():

Assume you correctly set this breakpoint and performed a run with correct command line arguments.
State how to print the value of the 2\textsuperscript{nd} argument passed to displayDiamond() 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 ( a = 9212; a >= 154; --a )
{
    x = a - 445;
}
```

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 % 9212;
```

SPARC assembly

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

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

Optimized version of above SPARC assembly