#1. Assume a signed 32-bit int variable $x$ has some non-zero value stored in it. Write 5 different single statements (loops and function calls are not allowed) in C to set the value in variable $x$ to zero.

__________________________   __________________________ 
__________________________   __________________________ 
__________________________

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

a)    set 0xABCD1234, %o1
      sra %o1, 20, %o1

Value in %o1 at this point is 0x__________________________

b)    set 0xABCD1234, %o1
      set 0x1234ABCD, %o2
      xor %o1, %o2, %o1

Value in %o1 at this point is 0x__________________________

c)    set 0xABCD1234, %o1
      set 0x1234ABCD, %o2
      and %o1, %o2, %o1

Value in %o1 at this point is 0x__________________________

#3. What gdb command should you use to print the contents of a memory location where the address of the memory location in question is the first argument to a function you are currently stepping through? Don't worry about what function it is, you do not have access to the source, the function may have been written in assembly. Just generally think of stepping through a function call and you want to print the contents of the memory location pointed to by the first argument.

You can assume the 1st argument is a pointer to a char. Print the string in memory pointed to by this argument.

______________________________________

(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
x = 8888;

while ( x >= 55 )
{
    a = x - 1234;
    --x;
}
```

SPARC assembly

```assembly
/* x is mapped to %12 */
/* a is mapped to %14 */
```

---

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

C

```c
x = x % 6543;
```

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