

Signature \_\_\_\_\_

Name \_\_\_\_\_

cs30x \_\_\_\_\_

Student ID \_\_\_\_\_

Score: \_\_\_\_\_

**Quiz 2**  
**CSE 30**  
**Fall 2008**

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

```
char Yahoo[] = "Hack Me!";  
short stuff = -37;
```

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

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

Value in %o1 at this point is **0x**\_\_\_\_\_

b)        set  0xFACEBEAD, %o1  
          set  0x3C3CACDC, %o2  
          xor  %o1, %o2, %o1

Value in %o1 at this point is **0x**\_\_\_\_\_

c)        set  0xFACEBEAD, %o1  
          set  0x3C3CACDC, %o2  
          and  %o1, %o2, %o1

Value in %o1 at this point is **0x**\_\_\_\_\_

#3. Assume you run gdb on pal.

State how to set a breakpoint at the entry point in displaySquare():

Assume you correctly set this breakpoint and performed a run with correct command line arguments.

State how to print the value of the 2<sup>nd</sup> argument passed to displaySquare() 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**

```
a = 5678;

while ( a > 181 )
{
    x = a - 37;
    a = a + 83;
}
```

**SPARC 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**

```
x = x / 9876;
```

**SPARC assembly**

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

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

**Optimized version of above SPARC assembly**