

Signature \_\_\_\_\_

**Quiz 5**  
**CSE 131**  
**Winter 2010**

Name \_\_\_\_\_

Login name \_\_\_\_\_

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

1. Use virtual register notation for each of the following.

Change the following into three instructions which are most likely a time improvement over the single instruction when it comes to actual code generation.

`r4 = r2 * 514`

_____
_____
_____

What term describes this particular kind of peephole optimization?
--

Change the following into another single instruction which is most likely a time improvement over the current instruction when it comes to actual code generation.

`r1 = 50 / 2`

_____
-------

What term describes this particular kind of peephole optimization?
--

Change the following into three instructions which are most likely a time improvement over the single instruction when it comes to actual code generation. `x` represents a memory location.

`r3 = r5 + r6`  
`x = r3`  
`r3 = x`  
`r6 = r7 + r3`  
`x = r6`

_____
_____
_____

What term describes this particular kind of peephole optimization?
--

Given the following code where ??? may represent a number of different parameter passing modes:

<pre>int global = 5;  void foo( int ??? param ) {     param = 10;      cout &lt;&lt; global &lt;&lt; endl; }</pre>
--

<pre>int main() {     foo( global );      cout &lt;&lt; global &lt;&lt; endl;      return 0; }</pre>
--

What values do you expect to be printed if the parameter passing mode is call-by-reference? \_\_\_\_\_  
What values do you expect to be printed if the parameter passing mode is call-by-value/result? \_\_\_\_\_  
What values do you expect to be printed if the parameter passing mode is call-by-value? \_\_\_\_\_

## 2. Given the following Reduced-C code fragment:

```
function : int foo( int x, int & y, int * z ) { /* Body of code not important for this question */ }

function : int main()
{
    int a = 8675309;
    int b;
    int c = a;

    b = foo( a, b, &c );

    return a;
}
```

Complete the SPARC Assembly language statements that might be emitted by a compliant Reduced-C compiler from this quarter for function main(). Allocate, store, and access all local variables on the Stack. See comments.

```

.section _____

.global _____
.align 4

_____:
set    _____, %g1
save  _____, %g1, _____

/* Initialize the local variables that have explicit initialization in this stack frame */
set    _____, %o0
st     %o0, _____          ! int a = 8675309;
ld     _____, %o0
st     %o0, _____          ! int c = a;

/* Set up the 3 actual arguments to foo() */
_____ _____, %o0 ! large blank can be one or two operands
_____ _____, %o1
_____ _____, %o2

call   foo                ! Call function foo()

_____

st     _____, [%fp - 16]    ! Save return value into local temp1

/* Copy saved return value stored in temp1 into local var b */
_____ [%fp - 16], _____

_____ _____, _____    ! b = foo( ... );

/* return a; */

ld     _____, _____    ! return a;

_____

_____

MAIN_SAVE = -(92 + _____) _____ ! Save space for 3 local vars + 1 temp
```

What question would you like to see on the Final?