

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

# Quiz 5 CSE 131

Name \_\_\_\_\_

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

## Spring 2010

Signature \_\_\_\_\_

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.  $x$  represents a memory location.

```

r5 = 42
x = r3
r7 = r6 + r5
x = r7
r5 = r1 - r2

```

---



---



---

What terms describes these particular kinds of peephole optimizations? There are at least two.

1)

2)

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

```
r1 = r3 * 254
```

---



---



---

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 = 99 % 3
```

---

What term describes this particular kind of peephole optimization?

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

```

int global = 13;

void foo( int ??? param )
{
    param = 4;

    cout << global << endl;
}

```

```

int main()
{
    foo( global );

    cout << global << endl;

    return 0;
}

```

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

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? \_\_\_\_\_

## 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;
    int b = 420024;
    int c = b;

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

    return b;
}
```

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

    _____, %g1, _____

    /* Initialize the local variables that have explicit initialization in this stack frame */

    _____, %o0

    st    %o0, _____          ! int b = 420024;

    _____, %o0

    st    %o0, _____          ! int c = b;

    /* Set up the 3 actual arguments to foo() */

    _____, %o0 ! large blank can be one or two operands
    _____, %o1
    _____, %o2

    _____ foo                ! Call function foo()

    _____

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

    /* Copy saved return value stored in temp1 into local var a */

    _____ [%fp - 16], _____

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

    /* return b; */

    ld    _____, _____    ! return b;

    _____

    _____

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

```

What question would you like to see on the Final?