

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

# Quiz 4 CSE 131 Winter 2012

Name \_\_\_\_\_

Signature \_\_\_\_\_

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

## 1. Project II Code Gen:

What is the output of the following Reduced-C program:

```
function : int foo( int x, int & y )
{
  int z = y;

  ++x;
  ++y;
  --z;

  cout << x << endl;
  cout << y << endl;
  cout << z << endl;

  return y++;
}

function : int main( )
{
  int a = 3;
  int b = 9;
  int c;

  c = foo( a, b );

  cout << a << endl;
  cout << b << endl;
  cout << c << endl;

  return 0;
}
```

Parameter passing mode for **x** \_\_\_\_\_

Parameter passing mode for **y** \_\_\_\_\_

Output

Fill in the blanks below to simulate the above in C. Basically what is really happening under the code.

```
int
foo( _____ x, _____ y )
{
  int z = _____;

  ++ _____;
  ++ _____;

  --z;

  printf( "%d\n", _____ );
  printf( "%d\n", _____ );
  printf( "%d\n", z );

  return _____;
}
```

```
int
main( )
{
  int a = 3;
  int b = 9;
  int c;

  c = foo( _____, _____ );

  printf( "%d\n", a );
  printf( "%d\n", b );
  printf( "%d\n", c );

  return 0;
}
```

Parameter passing mode for **x** \_\_\_\_\_

Parameter passing mode for **y** \_\_\_\_\_

2. 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? \_\_\_\_\_

call-by-reference? \_\_\_\_\_

call-by-value/result? \_\_\_\_\_

What C/C++ compiler option should you use to produce a .s file from a .c file? \_\_\_\_\_

What C/C++ compiler option should you use to produce a .o file from a .c file? \_\_\_\_\_

Assume local int \* variables a and b are allocated space in a function's stack frame at memory locations

```
int * a    %fp-4
int * b    %fp-8
```

Complete the SPARC assembly instructions for the line

```
b = a--;
```

that a Reduced-C compiler from this quarter might emit.

You can assume all the initializations of the local variables have been performed. Just emit the code to perform the expression on the right side of the assignment statement and assign the result into b.

We will need to use a temporary or two on the stack, so we will use location %fp-12 for tmp1 and %fp-16 for tmp2.

Follow the basic ld/ld/compute/st model.

```
ld      [_____], %o0
_____ %o0, [%fp - 12]      ! tmp1 = a

ld      [_____], %o0
_____ _____, %o1
sub     %o0, %o1, %o0
_____ %o0, [_____]      ! tmp2 = a - ?

_____ [_____], %o0
_____ %o0, [_____]      ! a = tmp2

_____ [_____], %o0
st      %o0, [_____]      ! b = tmp1
```