Login name _____ **Quiz 4** **Name** _____

CSE 131

Signature _____ **Spring 2009** **Student ID** _____

1. Project II Code Gen – Phase II.3:
What is the output of the following Reduced-C program:

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

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

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

  return x++;
}

function : int main( )
{
  int a = 4;
  int b = 8;
  int c;

  c = foo( a, b );

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

  return 0;
}
```

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 = 4;
  int b = 8;
  int c;

  c = foo( __________, __________ );

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

  return 0;
}
```

(over)

2. Name a typical operation that occurs at each stage of a subroutine calling sequence as discussed in class. There are usually several operations at each stage. Just name one for each stage.

Pre-Call (in caller):

_____

Prologue (in callee):

_____

Epilogue (in callee):

_____

Post-Return (in caller):

_____

---

3. Using the Rt-Lt Rule, write the C function prototype (declaration) for the following:

fubar is a function that takes two arguments – a pointer to a char and a pointer to a pointer to a float – and returns a pointer to an array of 17 elements where each element is a pointer to an array of 6 elements where each element is a pointer to a pointer to an int.

---

4. 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.

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


ld      [________________], %o0
_______ 1, %o1
sub     %o0, %o1, %o0
_______ %o0, [________________]   ! tmp2 = a - 1


_______ [________________], %o0
_______ %o0, [________________]   ! a = tmp2


_______ [________________], %o0
st      %o0, [________________]   ! b = tmp1
```