

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

**Quiz 4**  
**CSE 131**

Name \_\_\_\_\_

Signature \_\_\_\_\_

**Winter 2011**

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

1. Use the options below to fill in the blanks

- |             |             |            |         |         |
|-------------|-------------|------------|---------|---------|
| 1) positive | 4) negative | 7) %o0-%o5 | 10) -68 | 13) %o0 |
| 2) %fp      | 5) %sp      | 8) +68     | 11) -4  | 14) %i0 |
| 3) %pc      | 6) +4       | 9) %i0-%i5 | 12) +0  | 15) %g0 |

In the SPARC architecture, local variables are stored and accessed with a \_\_\_\_\_ offset from the current \_\_\_\_\_ register. Actual arguments are passed to functions by the caller in the \_\_\_\_\_ registers and are retrieved by the callee in the \_\_\_\_\_ registers. They should be stored in the current/callee's stack frame's formal parameter area with the first argument stored at offset \_\_\_\_\_ from the current \_\_\_\_\_ register. When the callee is ready to return, the return value is put in the \_\_\_\_\_ register. Back in the caller, the return value is retrieved from the \_\_\_\_\_ register.

---

Using the Right-Left Rule, write the C function prototype (declaration) for the following:

fubar is a function that takes two arguments – a pointer to a pointer to an int and a pointer to a struct bar – and returns a pointer to a function that takes no arguments and returns a pointer to an array of 4 elements where each element is a pointer to a pointer to a double.

---

Pick one of the following to answer the questions below related to most calling conventions.

- |   |   |                       |                      |                      |
|---|---|-----------------------|----------------------|----------------------|
| 1) Pre-Call (Caller)                                    | 2) call/jsr   | 3) Post-Call (Caller) | 4) Prologue (Callee) | 5) Epilogue (Callee) |
| _____ Retrieves return value from return value location | _____ Saves %pc into the return address location    |                       |                      |                      |
| _____ Stores return value into return value location    | _____ Retrieves saved return address for return/rti |                       |                      |                      |
| _____ Allocates space for local variables               | _____ Performs initialization of local variables    |                       |                      |                      |
| _____ Copies actual arguments into argument space       | _____ Saves registers in callee-save scheme         |                       |                      |                      |

---

Pick one of the following to answer the questions below:

- 1) Compile time      2) Run time

Method overloading is resolved at \_\_\_\_\_.

Method overriding is resolved at \_\_\_\_\_.

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

```
int a; // allocated at %fp-4
int b; // allocated at %fp-8
```

Complete the SPARC assembly instructions for the line

```
a = b++;
```

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 a.

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 = b

ld      [_____], %o0
_____ 1, %o1
add     %o0, %o1, %o0
_____ %o0, [_____]      ! tmp2 = b + 1

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

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

Given the following variable definitions

```
int x = 420; // initialized global variable
static int y; // uninitialized static variable
```

write the SPARC assembly code which should be generated to properly allocate space for each along with their initial values and alignment and to ensure proper access/visibility to these variables if another file is linked to this code's object file.

```
.section _____
.align _____
x: _____ 420
_____ x
.section _____
.align _____
y: _____ _____
```