

Login name _____

Quiz 4
CSE 131B
Winter 2006

Name _____

Signature _____

Student ID _____

1. Write the oberon program that corresponds to the following SPARC assembly output.

```
.section    ".text"
.align     4

.global    foo
foo:
    set     foo.SIZE, %g6
    save   %sp, %g6, %sp

    ! Integer parameter "x" is in %i0
    add    %i0, %i0, %i0

    ret
    restore

foo.SIZE = -(92 + 0) & -8

!-----
.section    ".rodata"
.align     4
ifmt: .asciz "%d"

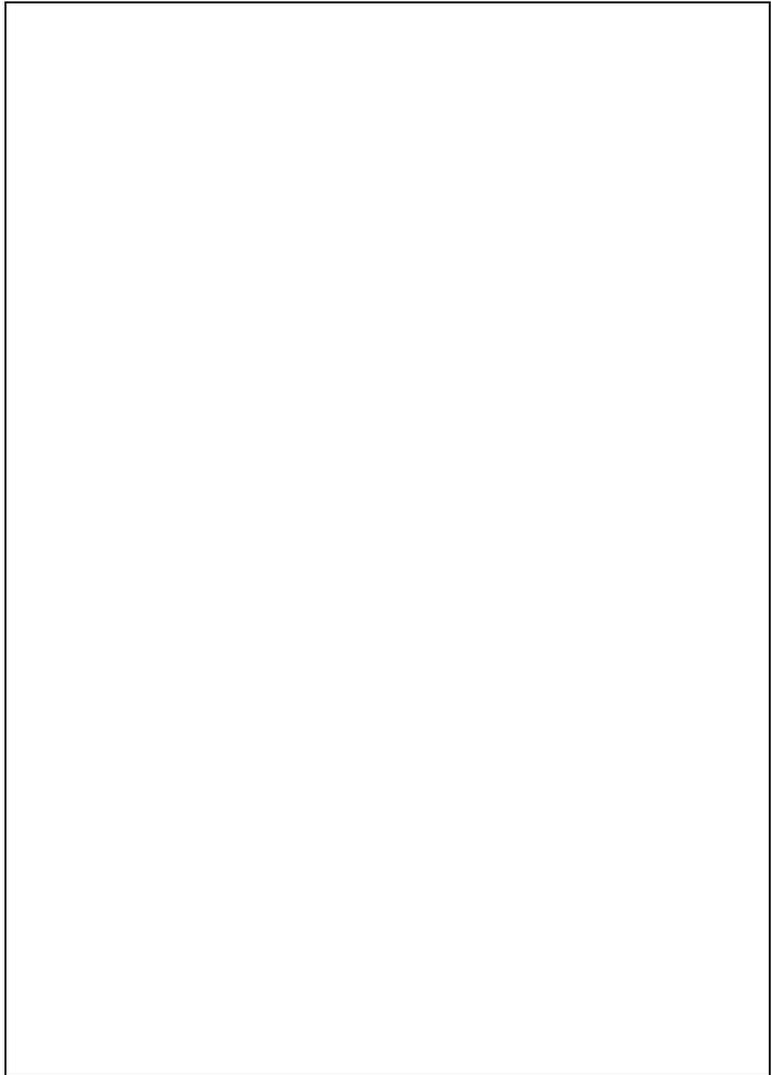
.section    ".text"
.align     4

.global    main
main:
    save   %sp, -96, %sp

    set    420, %o0
    call   foo
    nop

    mov    %o0, %o1
    set    ifmt, %o0
    call   printf
    nop

    ret
    restore
```



2. In the SPARC architecture, local variables are accessed with a _____ offset from the current activation record's base/index register _____.

3. What is the name of the research project Herb Sutter is working on that he talked about at the end of his talk?

4. Name a typical operation that occurs at each stage of a subroutine calling sequence. 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)

5. In C/C++, which are semantically correct function calls based on the function prototype for **foo()** and the definition of **a** below.

A) No compile error (semantically incorrect)

B) Compile error (semantically correct)

```
void foo( int array[] );
```

```
int a[10];
```

foo(a); _____ foo(&a[0]); _____ foo(a[0]); _____

foo(*a); _____ foo(*(a + 3)); _____ foo(&*a); _____

foo(*&a[0]); _____ foo(a + 3); _____

6. Given the following Oberon procedure, complete the equivalent generated SPARC assembly. **NOTE: Only registers %i0, %i1, and %i0 are available to use.** There should be a pair of ld/st instr. for each Oberon instr.

```
PROCEDURE swap2(VAR a:INTEGER; VAR b:INTEGER);
  VAR temp:INTEGER;
BEGIN
  temp := a; (* 1 *)
  a := b;   (* 2 *)
  b := temp; (* 3 *)
END swap2;
```

```

.global swap2
.section ".text"
swap2:
  save %sp, -(92 + _____) & -8, %sp ! alloc. temp
/* Stmt 1 */

/* Stmt 2 */

/* Stmt 3 */

ret
restore
```