

Login name _____

Quiz 4

Name _____

CSE 131B

Signature _____

Spring 2005

Student ID _____

1. State which calling convention / parameter passing mode is being used and what gets printed:

```

int x = 101;
void fool( int a, int *b ) {
    a = -9;
    *b = 7;
}
int main( void ) {
    int y = 66;
    fool( x, &y );
    printf( "x = %d; y = %d\n", x, y );
    return 0;
}

```

Parameter passing mode for **a** _____

Parameter passing mode for **b** _____

Output: x = _____; y = _____

Oberon

```

VAR x : INTEGER;
VAR y : INTEGER;
PROCEDURE fool ( VAR a : INTEGER; b : INTEGER );
BEGIN
    a := 14;
    b := 7;
END fool;
BEGIN
    x := 33; y := 55;
    fool( x, y );
    OUTPUT "x = ", x, "; y = ", y, "\n";
END.

```

Parameter passing mode for **a** _____

Parameter passing mode for **b** _____

Output: x = _____; y = _____

Fill in the blanks of the equivalent C program to simulate the above Oberon parameter passing modes (that exposes what the compiler is actually doing to implement these parameter passing modes):

```

int x, y;
void fool( _____ a, _____ b ) {
    _____ = 14;
    _____ = 7;
}
int main( void ) {
    x = 33; y = 55;
    fool( _____ , _____ );
    printf( "x = %d; y = %d\n", x, y );
    return 0;
}

```

Pick of one the following letters to answer the questions below.

A) Pre-Call

C) Prologue

B) Post-Return

D) Epilogue

_____ Where local variable space is allocated

_____ Where parameter space is allocated

_____ Retrieve return value from %o0 in SPARC subroutine

_____ Performs initialization of local variables

_____ Saves the return address

_____ Restores caller-save registers

What would a bare bones `oberon.s` file look like after compiling an Oberon program that has nothing except

```
BEGIN
```

```
    OUTPUT 17;
```

```
END.
```