

**1. mov vs lea - describe the difference between the following:**

```
movl (%rdx), %rax  
leal (%rdx), %rax
```

**2. Invalid `mov` Instructions - Explain why these instructions would not be found in an assembly program.**

- a) `movl %eax, %rdx`
- b) `movb %di, 8(%rdx)`
- c) `movq (%rsi), 8(%rbp)`
- d) `movw $0xFF, (%eax)`

**3. What would be the corresponding instruction to move 64 bits of data from register `%rax` to register `%rcx`?**

4. Operand Form Practice (see page 181 in textbook)

Assume the following values are stored in the indicated registers/memory addresses.

<u>Address</u>	<u>Value</u>	<u>Register</u>	<u>Value</u>
0x104	0x34	%rax	0x104
0x108	0xCC	%rcx	0x5
0x10C	0x19	%rdx	0x3
0x110	0x42	%rbx	0x4

Fill in the table for the indicated operands:

<u>Operand</u>	<u>Value</u>	<u>Operand</u>	<u>Value</u>
\$0x110	_____	3(%rax, %rcx)	_____
%rax	_____	256(, %rbx, 2)	_____
0x110	_____	(%rax, %rbx, 2)	_____
(%rax)	_____		
8(%rax)	_____		
(%rax, %rbx)	_____		

5. Condition Codes and Jumps - Assume the addresses and registers are in the same state as in the previous problem. Does the following code result in a jump to .L2?

```
leaq (%rax, %rbx), %rdi
cmpq $0x100, %rdi
jg .L2
```

6. Which of the functions cool1, cool2, or cool3 would compile into this assembly code?

```
    movl %esi, %eax
    cmpl %eax, %edx
    jge .L4
    movl %edx, %eax
.L4:
    ret

int cool1(int a, int b) {
    if ( b < a )
        return b;
    else
        return a;
}

int cool2(int a, int b) {
    if ( a < b )
        return a;
    else
        return b;
}

int cool3(int a, int b) {
    unsigned ub = (unsigned) b;
    if ( ub < a )
        return a;
    else
        return ub;
}
```