

Name:

UID:

1. mov vs lea?

Describe the difference between the following:

```
movq (%rdx), %rax  
leaq (%rdx), %rax
```

2. Instruction Junction

- What would be the corresponding instruction to move 64 bits of data from register %rax to register %rcx?
- What would be the corresponding instruction to move 64 bits of data from the memory location stored in register %rax to register %rcx?

3. What was Compiled?

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

```
movl %esi, %eax  
cmpl %eax, %edi  
jge .L4  
movl %edi, %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;  
}
```

4. Operand Form Practice

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

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

Fill in the table for the indicated operands:

Operand	Value	Value (if lea)
\$0x110		
%rax		
0x110		
(%rax)		
8(%rax)		
(%rax, %rbx)		
3(%rax, %rcx)		
256(, %rbx, 2)		
(%rax, %rbx, 2)		

5. Disassembled Function

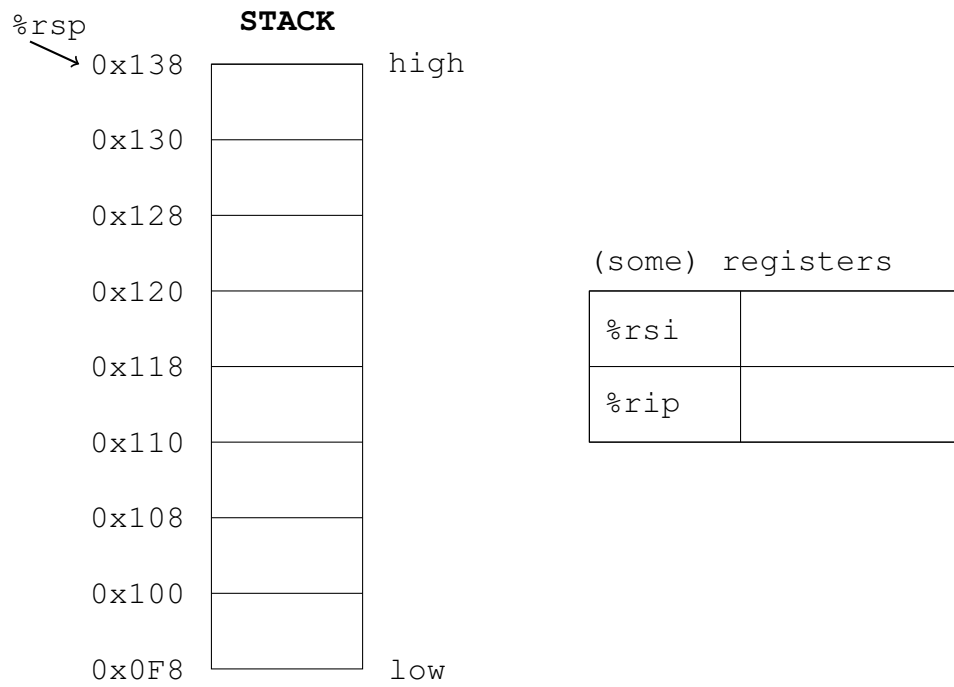
Consider the following disassembled function:

```

000000000040102b <phase_2>:
40102b: 55                push %rbp
40102c: 53                push %rbx
40102d: 48 83 ec 28       sub $0x28,%rsp
401031: 48 89 e6          mov %rsp,%rsi
401034: e8 e3 03 00 00    callq 40141c <read_six_numbers>
401039: 83 3c 24 01       cmpl $0x1, (%rsp)

```

- (a) Assume `%rsp` initially has a value of `0x138`. Draw the stack (see example diagram below) for the execution of `<phase_2>`, updating the stack and register values as necessary after each line.



- (b) Right after the `callq` instruction has been executed, what are the values of `%rsp`, `%rsi`, and `%rip`?

Bonus: 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)`

Bonus: Condition Codes and Jumps

Assume the addresses and registers are in the same state as in Problem 4. Does the following code result in a jump to `.L2`?

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