

## 1. Data Lab Practice (from codesignal.com)

Write a function that, given a number  $n$ , returns another number where the  $k^{\text{th}}$  bit from the right is set to 0.

Examples:

killKthBit(37, 3) = 33 because  $37_{10} = 100\mathbf{1}01_2 \leadsto 100\mathbf{0}01_2 = 33_{10}$

`killKthBit(37, 4) = 37` because the 4<sup>th</sup> bit from the right is already 0.

```
int killKthBit(int n, int k) {
```

}

2. `mov` vs `lea` - describe the difference between the following:

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

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

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

4.

```
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;
}

```

Which of the functions would compile into this assembly code:

```

    movl %esi, %eax
    cmpl %eax, %edi
    jge .L4
    movl %edi, %eax
.L4:  ret

```

## 5. 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)	_____		