1. Write a function that, given a number n, returns another number where the k^{th} bit from the right is set to to 0.
Examples:
`killKthBit(37, 3) = 33` because $37_{10} = 100101_2 \sim> 100001_2 = 33_{10}$
`killKthBit(37, 4) = 37` because the 4^{th} bit 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. 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)
```

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

5. Operand Form Practice (see page 181 in textbook)
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	Operand	Value
$0x110	_____	3(%rax, %rcx)	_____
%rax	_____	256(, %rbx, 2)	_____
0x110	_____	(%rax, %rbx, 2)	_____
(%rax)	_____		
8(%rax)	_____		
(%rax, %rbx)	_____		

6. Condition Codes and Jumps - Assume the addresses and registers are in the same state as in Problem 6. Does the following code result in a jump to `.L2`?

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