1. Data Lab Practice (from codesignal.com)

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 $37_{10} = 100101_2$ ~> $100001_2 = 33_{10}$
killKthBit(37, 4) = 37 because the 4th bit from the right is already 0.

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

      return n & (~(1 << (k - 1)));

}
```

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

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

Movq will dereference to memory, while leaq will not. Therefore, it makes sense that registers accessed by movq contain addresses in memory while registers accessed by leaq contain values that can immediately be used without dereference, such as integers.

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

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

cool2 I think.

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	_____272_____	3(%rax, %rcx)	_____25_____
%rax	______0x104______	256(, %rbx, 2)	______204______
0x110	_____66_____	(%rax, %rbx, 2)	_____25_____
(%rax)	______52______		
8(%rax)	_____25_____		
(%rax, %rbx)	_____204_____		