CSCI-UA.0201-003
Computer Systems Organization
Midterm Exam Fall 2015 (time: 60 minutes)

Last name:           First name:

Notes:
• If you perceive any ambiguity in any of the questions, state your assumptions clearly.
• Questions vary in difficulty; it is strongly recommended that you do not spend too much time on any one question.

1. [2 points] What would have happened if we didn’t have linkers? State 2 consequences.
   -
   -

2. [5 points] Fill in the blanks in the following table, assume the instructions are executed sequentially (i.e. instruction 1 executed, then instruction 2, then instruction 3. Assume rax and rbx are holding signed numbers.

<table>
<thead>
<tr>
<th>instruction#</th>
<th>instruction</th>
<th>rax</th>
<th>rbx</th>
<th>CF</th>
<th>ZF</th>
<th>SF</th>
<th>OF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initially</td>
<td></td>
<td>0xFFFFFFFF</td>
<td>0x00000001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>addq %rbx, %rax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>testq %rbx, %rax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>cmpq %rbx, %rax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. [5 points] For the following assembly code and its corresponding C code, fill-in the blanks in the C code assuming x will go into %rbx and y will go into %rax;

```assembly
cmpq %rax, %rbx                      x, y; /* declaration of x and y;
ja L1                                     if ( )
subq, %rbx, %rax                      {                                     }
jmp L2                                     
L1:      addq $10, %rax  }                         
L2:      addq %rbx, %rax  }                         
else                                     }
                     }                         
```

4. [2 points] Can the carry flag (CF) and the overflow flag (OF) be both 1 at the same time? If yes, give an example of an operation that does this (no need for assembly code, just describe the operation). If not, explain why not.

5. [2 points] State two reasons for why do we need an assembler and not making the compiler generate the binary presentation right away.

- 
- 

6. Suppose \( x \) is an integer (i.e. 4 bytes). We want to test whether the most significant bit of \( x \) is 1 or not (i.e. the left most bit), so we wrote the expression:

\[
\text{if}( \ (x \ & \ \text{mask}) \ != \ 0) \]

a. [1 point] What is the value of mask, both in binary and hexadecimal?

b. [2 points] Which of the following expressions generate correct mask? Circle ALL correct answers. There may be more than one correct answer, or there may be none!

- \( 0x1FFFFFFF \ll 3 \)
- \( 0x1FFFFFFF \ll 2 \)
- two’s complement of \( 0xFFFFFFFF \)
- two’s complement of \( -2 \)

c. [1 point] Please give the expression that sets the most significant bit of \( x \) to 1 and leave all the other bits unchanged.