Name: ____

Intro. to Computer Architecture	**Homework 5**
CSE 240 Autumn 2006	**Due: Wed. 25 October 2006**

Write your answers on these pages. Additional pages may be attached (with staple) if necessary. Please ensure that your answers are legible and *show your work.* Write your name at the top of each page. Due at the ***beginning of class***.

Note: This handout is **not** the entire assignment; this is **only** for Problems 3 and 4. Problems 1, 2 and 5 are to be turned in electronically. See the web page for homework 5 (linked off the course schedule) for the rest of this assignment.

3. **Debugging LC-3 Assembly Code I.** Consider the following LC-3 assembly code that's supposed to add up a list of numbers in memory. The memory location of the first number is given in R2, and the total number of numbers that need to be added is given in R3 (*i.e.*, R2 and R3 serve as input to this code). The resulting sum is placed in R5. The numbers to be added reside in consecutive memory locations. So if R2 == x4000 and R3 == 3, the numbers to be added are in x4000, x4001 and x4002.

   But there's a bug somewhere. Download the LC-3 assembly code from the course web site and run it on the simulator. **Hint:** Set a breakpoint on the branch; decide in your head whether the branch should be taken or not; then use the simulator's `step` command to follow the branch and see what actually happens; if the right thing happened, hit `continue` to reach the breakpoint in the next iteration of the loop; repeat. Determine what the bug is, then answer the questions below.

   (a) One instruction in the program is not quite right. Which one is it and to what should it be changed?

Label	Instruction	Modified Instruction
START	AND R5,R5,#0	
LOOP	ADD R3,R3,#-1	
	LDR R4,R2,#0	
	ADD R2,R2,#1	
	ADD R5,R4,R5	
	ADD R3,R3,#0	
	BRzp LOOP	
QUIT	HALT	

   (b) In one sentence, what was wrong with the code?

4. **Debugging LC-3 Assembly Code II.** Consider the following LC-3 assembly code that's supposed to find the largest value in a list of non-negative numbers and put that value in R5. The memory location of the beginning of the list is in R2, and the end of the list of numbers is signified by a negative value. That is, if R2 == x4000 and the contents of memory are as below, then the value 2 should be placed in R5.

Address	Value
x4000	2
x4001	1
x4002	-1

But there's a bug that prevents this from happening. Download the LC-3 assembly code from the course web page and run it on the simulator. **Hint:** Set a breakpoint at LOOP, and use the simulator's `continue` command to count the number of iterations. Determine what you think should be the "last" iteration and find the place where the loop should end, but doesn't. Determine what the bug is and then answer the questions below.

(a) How many times does the loop iterate in the provided code?

(b) To fix the code, you need to insert an instruction somewhere. Where would you do this, and what instruction would it be? Write your answer in one of the blanks below.

Label	Instruction
START	AND R5,R5,#0
LOOP	LDR R3,R2,#0
	ADD R2,R2,#1
	NOT R4,R5
	ADD R4,R4,#1
	ADD R4,R3,R4
	BRn LOOP
	ADD R5,R3,#0
	BRnzp LOOP
QUIT	HALT

(c) In one sentence, what was wrong with the code?