

# EECS 370 Winter 2009

## Homework 1

**Assigned:** Tuesday, January 13, 2009

**Due:** Tuesday, January 27, 2009 (in class)

Name: \_\_\_\_\_ Uniqname: \_\_\_\_\_

### Instructions:

1. Please write your name and uniqname in the spaces provided above. *Attach this cover sheet* to your completed solutions to the problems listed and turn them in at class on the due date. Submissions without a completed, attached cover sheet cannot be graded.
2. Your answers should be neat, clear, and concise. Computer-written work is recommended (but not required). Show all your work, and state any special or non-obvious assumptions you make.
3. You may discuss your solution methods or your answers with other students, but the solutions you submit must be your own.

Scores:

<b>Problem #</b>	<b>Points</b>
1	/6
2	/8
3	/4
4	/12
<b>Total</b>	<b>/30</b>

**Problem 1** (6 points)

Convert the following decimal values to both binary and hexadecimal representations.

Example:  $132_{10} == 10000100_2 == 84_{16}$ .

$196_{10}$

$2807_{10}$

$49_{10}$

**Problem 2** (8 points)

Given the following MIPS assembly code:

```
    add $1, $0, $0
    addi $2, $0, 50
lp:  andi $3, $2, 1
     bne $3, $0, 1
     add $1, $1, $2
     addi $2, $2, -1
     bne $2, $0, lp
```

How many instructions would be executed by a MIPS processor running this code? Also provide a brief one or two sentence explanation of what the code is doing.

**Problem 3** (4 points)

Assume a 64-bit wide RISC machine with 64 registers and 200 distinct instructions. Its branch instruction compares two registers for equality, and uses the 2's complement offset field to compute the PC relative target. What is the branch target range?

**Problem 4** (12 points)

Most computers represent signed numbers in 2's-complement. Given a 32-bit number, in 2's-complement, write LC-2K8 assembly code that calculates the absolute value. Assume that the number is already stored in register 1 and the absolute value is to be placed in register 2. The original number in register 1 does not need to be preserved. Try to do this in less than a dozen lines of code.