UNCA CSCI 255
Exam 2 Fall 2012
15 November, 2012

This is a closed book and closed notes exam. It is to be turned in by 10:40 AM. Calculators, PDA's, cell phones, and any other electronic or communication devices may not be used during this exam.
If you want partial credit for imperfect answers, explain the reason for your answer!

Name:_____

Problem 1 (4 points) Decimal to two's complement conversion
Convert the following two signed decimal numbers into eight-bit two's complement representation. Some of these numbers may be outside the range of representation for eight-bit two's complement numbers. Write "out-of-range" for those cases.

-24	**36**

Problem 2 (4 points) Two's complement to decimal conversion
Convert the following four six-bit two's complement numbers into signed decimal representation.

101010	**010101**

Problem 3 (6 points) Memories
A 32 MB memory has a 16-bit word size. How many words are contained in this memory?

How many bits are required to address the words of this memory?

Problem 4 (6 points) Adding
Add the following three pairs of six-bit two's complement numbers **and indicate which additions result in an overflow by writing either "overflow" or "no overflow" in each box**.

001101 + 010101	011010 + 100110
110110 + 111010	

Problem 5 (10 points) Boolean expression to truth table
Complete the truth table below so that it corresponds to the following Boolean equation

X = A C + $\overline{A}$ B C

If you prefer that your inversions be primes, you can think of the equation as

X = A C + A' B C

Or, if you really like Java conditional expressions, you can go with

```
X = A && C || !A && B && C
```

A	B	C	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Problem 6 (15 points) Truth table to digital logic
Draw a logic circuit at the gate level that will implement the following truth table, where A, B, and C are inputs and Z is the single output.

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Problem 7 (5 points) Truth table to Boolean expression
Write a Boolean expression that corresponds to the truth table shown in Problem 6. You can build on your Problem 6 answer if that seems appropriate.

Problem 8 (12 points)
In the tables below the values of several registers and memory locations are given.

Registers	
W0	0x00FF
W1	0x0006
W2	0x0902
W3	0x0904
W4	0x0906

Memory	
0x0900	0x0005
0x0902	0x0007
0x0904	0x0908
0x0906	0x090A
0x0908	0xFFFF

For the following six PIC24 instructions, list any registers or memory locations that are modified by the instruction along with the value stored into the register or location. Assume that each instruction is executed using the values shown above. That is, the instruction executions are *not* sequential. You can give the new value in either decimal or hexadecimal.

`MOV #42,[W3]`
`MOV [W1+W2],W3`
`INC W1,W1`
`AND W4,#0xF,W3`
`MOV [W4++],W1`
`MOV [--W2],W0`

This is the 2nd Problem 8

Problem 8 (12 points)
In the left column there are six PIC24 instructions. In the right column write the 24 bits needed to encode each instruction in the PIC24 instruction set. (By the way, you should have determined the binary encoding of one of these numbers in Problem 1.)

add W2,W6,W10	_____
add W2,#6,W10	_____
mov PCH,WREG	_____
mov WREG,PCH	_____
mov W0,PCH	_____
mov #36,W13	_____

Setup for the remaining problems
In the remaining problem, similar to what you saw in lab, you are going to write short sections of PIC24 assembly code corresponding to C statements. In writing these code, assume that `x` and `z` are 16-bit integers, which could be declared in C as follows:

```
int   x ;
int   z ;
```

In PIC24 assembly code, they would be declared using the `.space` directive:

```
x:   .space  2
z:   .space  2
```

In each of the programs a short section of C code will be given. You are to translate that code into a PIC24 code. You are free to use `WREG` and any of the sixteen PIC24 working registers in your answers. In some instances you will also need to define labels.

Just to make sure you don't make any beginner errors in the excitement of working these problems, keep in mind that the following are **not** legal PIC24 instructions and should **not** be used in your solutions.

```
mov   #7,x        ;; no literal with move to SFR
add   x,W0        ;; no add of SFR to working register
add   x,z         ;; no add of two SFR's
```

Often you will need to load `x` or `z` into a register before using them in a calculation.

Problem 8 needs three instructions. Problems 9 to 11 require five to seven.

And here's a 3rd Problem 8!

Problem 8 (5 points)

```
x = 7 ;
z = 7 ;
```

Problem 9 (7 points)

```
z = 3*z + 1 ;
```

Problem 10 (7 points)

```
if (x == z && x > 0) {
    ++x ;
}
```

Problem 11 (7 points)

```
while (x > 0) {
    x = x - z ;
}
```