ECE/CS 250 Midterm Exam #1, Spring 2016

Name: _____

Duke students are bound by an academic integrity standard:

1. I will not lie, cheat, or steal in my academic endeavors, nor will I accept the actions of those who do.

2. I will conduct myself responsibly and honorably in all my activities as a Duke student.

Please sign your name below to acknowledge that you follow this standard:

---

1) [5 points] (a) Represent the base-10 numbers 11 and -4 in 7-bit 2s complement representation.

(b) [5 points] Add these two numbers together using 2s complement arithmetic. Do NOT invert the negative number and then add two positive numbers together. You must just directly add the two numbers. Show your work.

2) [2 points] Write 164_{10} in hexadecimal (base 16).

[8 points] The IEEE 754 floating point standard specifies that 32-bit floating point numbers have one sign bit, an 8-bit exponent (with a bias of 127), and a 23-bit significand (with an implicit "1"). What floating point number is represented by the following 32 bits?

0 1000001 01010000000000000000000

3) [5 points] (a) What is the difference between architecture and microarchitecture?

(b) [5 points] Which of the following issues are part of the instruction set architecture? Circle the issues that are part of the ISA.

- **The number of registers**
- **The processor's performance**
- **The processor's power consumption**
- **The memory addressing modes**
- **The types of opcodes**

(c) [5 points] Are the following statements true or false? Circle one for each.

MIPS is a RISC instruction set architecture. TRUE FALSE

A MIPS branch instruction has an immediate argument. TRUE FALSE

The heap holds dynamically allocated variables. TRUE FALSE

The unsigned binary representation of 1.75_{10} is 1.11_2. TRUE FALSE
(note: I'm NOT talking about floating point format)

If I add two 64-bit 2s complement integers, I can't overflow TRUE FALSE

4) [5 points] (a) Write one line of C code to dynamically allocate space for 32 ints on the heap.

[5 points] (b) At the end of the following snippet code, what are the values of A, B, C, D, and E? Put your results in the table to the right. Assume a 32-bit machine.

```
int x=3; // assume x is at address 1000
int* y = &x; // assume y is at address 1004
int** z = &y; // assume z is at address 1008
int numbers[100];
for (i=0; i<100; i++){
        numbers[i] = i;
}
int A = *(numbers + 2);
int B = *y + 5;
y = numbers;
int C = y[20];
int* D = *y;
int* E = *z;
```

A	
B	
C	
D	
E	

5) [20] Convert the following C code for the function f() into MIPS code. Use appropriate MIPS conventions for procedure calls, including the passing of arguments and return values, as well as the saving/restoring of registers. Assume that there are 2 argument registers ($a0-$a1), 2 return value registers ($v0-$v1), 3 general-purpose callee saved registers ($s0-$s2), and 3 general-purpose caller-saved registers ($t0-$t2). Assume $ra is callee-saved. The C code is obviously somewhat silly and unoptimized, but YOU MAY NOT OPTIMIZE IT -- you must translate it as is.

```
1: int f (int num){
2:     // set up stack
3:     int x = 0; // x must be in $s0
4:     int y = 1; // y must be in $t0
5:     if (num == 0) {
6:         y = num + 2;
7:     }else{
8:         y = num - y;
9:     }
10:    x = bar(x,y);
11:    y = x + y;
12:    return (y + 2);
13:    // clean up stack
14: }


 int bar (int arg) {
     // don't worry about bar()
 }
```

C lines	Lines of Assembly
1-2	
3-4	
5-9	
10	
11	
12	
13-14	