Quiz 2

Name:_____ uname:_____ lab section:_____

This is a closed-book, closed-note, closed-everything (including calculators) quiz. You will have 20 minutes. Be aware that there are different versions of the quiz.

1. Convert the following representations. If there is no representation for a given value, simply draw an "X" in the box. **[25, -3 per blank or wrong answer]**

Number	4-bit unsigned	4-bit signed-magnitude	4-bit 2's complement
-1	***X***	***1001***	***1111***
6	***0110***	***0110***	***0110***
-8	***X***	***X***	***1000***

2. What is printed by the following C++ program? **[20]**

```
#include <iostream>
using namespace std;

class bob
{
public:
    int local;
    bob();
    int cool(int a);
};

bob::bob()
{
    local=-4;
}
int bob::cool(int a)
{
    local=local*a+1;
    return(local*2);
}

main()
{
    bob a, b;
    int i=6;

    b.local=1;
    a.local=b.cool(i);
    cout << a.local << " " << b.local << endl;
}
```

3. ***14 7***

Comments: default constructor sets a.local and b.local to be –4. b.local is then made 1. In b.cool local (which is the same a b.local) is set to 1*6+1 or 7. The function returns 14. a.local is then set to 14.

Matlab: **[12]**
Show the matrix "x" which results from the following instructions:

1. x=7:2:12
   [7 9 11]

2. y=[2 5 7 1]
   x=y([2 3])
   [5 7]

4. Write a Matlab program which finds the sum of all values of a 2 by 2 matrix named bob. **[13]**

sum(sum(bob))
or
bob(1,1)+bob(1,2)+bob(2,1)+bob(2,2)
or
lots of other ways, but the above two are probably the best.

Recall the following:

Instruction	opcode [15:14]	memA[13:8]	B[7:0]
add	00	memA	memB
addi	01	memA	immediate
beq	10	memA	target
print	11	memA	unused

add: Mem[memA]=Mem[memA]+Mem[memB]
addi: Mem[memA]=Mem[MemA]+immediate
beq: if(Mem[memA]==0) PC=target
print: print Mem[memA] and halt.

So 0000 0000 0000 0000 says to add the byte at memory location 0 to itself and store the result in memory location zero.

5. Translate the following assembly program into machine code. Leave blank any unused space. **[30]**

```
add a, b
addi a, 1
print a
a: 5
b: -4
```

Comments
add in bytes 0 and 1.
addi in bytes 2 and 3
print in 4 and 5
a in 6
b in 7

* negative numbers in 2's complement
* execution starts at location 0
*

Address	Data
0	**00 000110**
1	**00000111**
2	**01 000110**
3	**00000001**
4	**11 000110**
5	**anything or nothing could go here**
6	**00000101**
7	**11111100**
8	
9	