CMSC B240 Computer Organization - Spring 2024
Lab Activity #2: Binary Number Operations

Student Name: ____

Student Name: ____

Question #1
Calculate the sums of the following **unsigned** binary integers. Think about how you can check that your answers are correct!

1.1) 1101 + 1001

1.2) 10101 + 110

Question #2
Calculate the sums of the following **8-bit 2's-complement** binary numbers; the results should also be expressed as 8-bit 2's-complement binary numbers:
2.1) 00010101 + 01001111

2.2) 00110000 + 01100001

2.3) 11110011 + 11111010

2.4) In which cases did overflow occur? How could you tell?

Question #3

In this question, you will see how you can use different techniques to perform subtraction using 8-bit 2's-complement binary numbers. In particular, we want to show how to calculate 27 - 15.

3.1) First, show how you can do this by performing binary subtraction on the two numbers, borrowing from columns to the left as needed:

```
    0 0 0 1 1 0 1 1
-   0 0 0 0 1 1 1 1
    _______________
```

Check that your result is correct!

3.2) Now calculate 27 - 15 as 27 + (-15) by first determining the binary representation of -15 and adding it to 27. You should get the same result as in Q3.1!

Question #4

Compute the following using bitwise operations:

4.1) (0110 OR 0000) AND 1111

4.2) NOT(1001 AND (1100 OR 0100))

Question #5

What would you expect to be printed by each of the following portions of Java code? Don't actually program it! Figure it out by hand!

5.1)
```
int x = 72;
int y = 40;
int z = 30;
System.out.println( (x & y) ^ z );
```

5.2)
```
int a = 31;
int b = -31;
System.out.println( a & b );
```

<u>Question #6</u>
Convert the following **unsigned** binary numbers to hexadecimal.

6.1) 10010011

6.2) 1101000110101111

6.3) 11110

<u>Question #7</u>
Convert the following hexadecimal numbers to binary.
7.1) xABC

7.2) x10

Question #8

Convert the following decimal numbers to hexadecimal. See if you can figure out how to do this without first converting to binary!

8.1) 82

8.2) 171

<u>Question #9</u>

Express the decimal value -78.5 using the IEEE 32-bit floating point representation, i.e. the representation used by the Java “float” datatype. Write your answer in hexadecimal.

Question #10
In addition to representing integers and floating point numbers in binary, we can also represent single characters such as ‘a’, ‘b’, ‘A’, ‘B’, ‘?’, etc.

A common representation is **ASCII**, in which each character has a unique 8-bit value.

A nice aspect of ASCII is that the letters of the alphabet are grouped together:
- Uppercase ‘A’ is represented using 01000001 = x41 = 65
- Uppercase ‘B’ is one greater than ‘A’, i.e. its representation is 01000010 = x42 = 66
- Uppercase ‘C’ is one greater than ‘B’, i.e. its representation is 01000011 = x43 = 67

and so on. The same applies to lowercase letters as well. You can find ASCII tables at the back of your textbook or online.

The Java programming language uses **Unicode** for representing characters, which is a 16-bit representation. ASCII representations are kept the same in Unicode, e.g. ‘A’ is x0041 in Unicode, and the letters of the alphabet are still grouped together, e.g. ‘B’ is x0042 and so on.

Based on this, what do you think is printed out by each of the following lines of Java code? Don’t actually program it! Figure it out by hand!

10.1)
```
System.out.println((char)('A' + 1));
```

10.2)
```
System.out.println((int)('A' + 1));
```

10.3)
```
System.out.println((int)('A' + 'B'));
```

10.4)
```
System.out.println((char)('M' - 5));
```