ISTA 350 2's Complement Worksheet Name:

Converting positive two's complement binary numbers to decimal: multiply each bit by the appropriate power of two and sum the resulting values. E.g.:

$$01100001_2 = \underline{\mathbf{0}} \cdot 2^7 + \underline{\mathbf{1}} \cdot 2^6 + \underline{\mathbf{1}} \cdot 2^5 + \underline{\mathbf{0}} \cdot 2^4 + \underline{\mathbf{0}} \cdot 2^3 + \underline{\mathbf{0}} \cdot 2^2 + \underline{\mathbf{0}} \cdot 2^1 + \underline{\mathbf{1}} \cdot 2^0$$

$$= 0 + 64 + 32 + 0 + 0 + 0 + 0 + 1 = 97_{10}$$

Convert the following positive two's complement binary numbers to decimal:

01001100_2 00011111_2

00100101_2 01110000_2

Adding two's complement binary numbers is done just like normal decimal addition. E.g.:

```
 01100001
+00001100
 --------
 01101101
```

Complete the following additions and check your work by converting all numbers to decimal. If the resulting number is negative, write 'overflow'.

```
 00011001          00111101          01000000
+00011101         +00011101         +01000000
 --------          --------          --------
```

Negating two's complement binary numbers: flip each bit and add 1. E.g.:

−(01100001) = 10011110 + 00000001 = 10011111

Convert the following negative two's complement binary numbers to decimal. I suggest first making them positive, then converting to decimal and placing a unary negation operator (minus sign) in front of them.

11001100_2 11111111_2

10100101_2 11110000_2

Complete the following additions and check your work by converting all numbers to decimal. Ignore any carry out of the leftmost digits. If the two operands have the same sign and the result has the opposite, write overflow.

```
 11111111
+11111111
---------
```

```
 10111101
+00011101
---------
```

```
 10000000
+10000000
---------
```

```
 00111101
+10011101
---------
```

```
 10000000
+01111111
---------
```

```
 11111111
+00000001
---------
```

```
 11000000
+11000000
---------
```

```
 01100011
+00100001
---------
```