HOMEWORK #2: SOLUTIONS/RUBRIC

PROBLEM #1:

a) +47

	rem:
47/2 = 23	1
23/2 = 11	1
11/2 = 5	1
5/2 = 2	1
2/2 = 1	0
1/2 = 0	1

↑

↳ 00101111_2

sign mag: 00101111_2 (+0.5)

1's comp: 00101111_2 (+0.5)

2's comp: 00101111_2 (+0.5)

b) -96

	rem:
96/2 = 48	0
48/2 = 24	0
24/2 = 12	0
12/2 = 6	0
6/2 = 3	0
3/2 = 1	1
1/2 = 0	1

↑

↳ 01100000

sign mag: 11100000 (+0.5)

1's comp: 10011111 (+0.5)

2's comp: 10100000 (+0.5)

c) -132

↳ signmag, 1's comp., and 2's comp. cannot be represented here in 8 bits since there will be an overflow

↳ only -127 to 127 can be rep. w/ 8 bits

(+0.5) for each conversion that is stated as not possible

↳ total (+1.5)

PROBLEM #2:

a) 01110 + 11101 (+1.5)

```
  01110
+ 11101
-------
 101011
+     1   (end-around carry)
-------
1's:  01100
↳ no overflow
```

```
 X01011
  01011
2's: 01011
↳ no overflow
```

b) 11101100 - 11010011 (+1.5)

↳ subtracting is same as adding comp.

```
  11101100 (-)
+ 00101100 (+)
----------
 100011000
+        1   (end-around carry)
----------
1's: 00011001
↳ no overflow
```

```
  11101100 (-)
+ 00101101 (+)
----------
 X00011001
2's: 00011001
↳ no overflow
```

(writing overflows is optional for all of Problem 2)

c) 100101 − 010010 (+1.5)

```
  100101 (-)          100101 (-)
+ 101101 (-)        + 101110 (-)
 -------             -------
 1010010 (-)        1010011 (+)   [leading 1 crossed out]
 ↪     1            2's: 010011
 -------
1's: 010011
```

↳ Overflow since two negative #'s are added and the answer is a positive

↳ also overflow

PROBLEM #3:

↳ You can verify that there is overflow when 2 neg. are added to get a pos. or if you add 2 pos. and get a negative

↳ Or if $C_n \cdot C_{n-1}$ have diff. signs

(+1) for either answer