Exercise 1:
(46)base 10 convert to hexadecimal

	Quotient	Remainder
46/2	23	0
23/2	11	1
11/2	5	1
5/2	2	1
2/2	1	0
1/2	0	1

(46)base 10 to binary: 101110
(101110)binary convert to hexadecimal
0010 1110 => (2E)hexadecimal

Exercise 2:
Find 2's complement of a signed number using 8-bits:
(-46) find 2's complement
1. convert 46 to binary: 101110
2. -46 = -101110

```
     00101110
1's: 11010001
2's:       +1
     11010010
```

Exercise 3:
Add the following two pairs of 8-bit numbers, which are already in their 2's complement hexadecimal forms if they are treated as signed integers.
Indicate whether your result is "right" or "wrong"
0x96 + ox97

```
  10010110 (0x96)
 +10010111 (0x97)
 100101101  for unsigned: wrong

  for signed: V = C ^ penultimate carry
              V = 1 ^ 0 = 1 : wrong
```

```
another example:
   1 <- penultimate carry
   00010101
  +01101111
   10000100
c=0
penultimate carry: 1


Exercise 4: calculate internet checksum
     1000  (8)
      1011  (11)
      1101  (13)
      1110  (14)

1. start with first two, then find the sum
    1000
  +  1011
   10011
      +1
     0100
  + 1101
   10001
      +1
    0010
  +  1110
   10000
      +1
    0001
1's:1110 => checksum

Exercise 5: calculate CRC-3
Message: 110010101
Pattern: 1001 (x^3 + 1 -> polynomial equivalent)
1. step #1: P(n)-1 : 110010101000    append zero's to the
message based on the pattern (P(n) - 1)
2. step #2: perform modulo-2 (XOR)

   1001 | 110010101000   Result: 110100001
```

```
          1001
            10110101000
            1001
             0100101000
            0000
               100101000
               1001
              00001000
              0000
               0001000
               0000
                  001000
                   0000
                    01000
                    0000
                     1000
                     1001
                      001
```

The message that gets send is: 110010101001

```
   1001 | 00110010101000      Result: 00
          0000
            0110010101000
           0000
             110010101000
```

* Review chapter 9 in Book 1 (diveintosystems)