

# Problem 5.1:

a)  $1_{10} = 0001_5$   $8_{10} = \cancel{4000}_5^{0013_5}$

b-complement:  $b=5, n=4$   $-1_{10} = 4444_5$   
 $-8 = 4432_5$

$$\begin{array}{r} 4444 \\ - 0001 \\ \hline 4443 \\ + 1 \\ \hline 4444 \end{array} \quad \begin{array}{r} 4444 \\ - 0013 \\ \hline 4431 \\ + 1 \\ \hline 4432 \end{array}$$

b)  $4444_5 + 4432_5 = 14431_5$   $4444_5 - 4431_5 = 0013_5 + 1 = 0014_5$   $0014_5 = 9_5$

# Problem 5.2:

a)  $273.15_{10} = 100010001.0010011_2$  ← shifting left 8 places  
 $8 + 127$  exponent  
 $135_{10} = 10000111_2$

2   273	0.15 × 2 = 0.30	0
2   136	0.3 × 2 = 0.6	0
2   68	0.6 × 2 = 1.2	1
2   34	0.2 × 2 = 0.4	0
2   17	0.4 × 2 = 0.8	0
2   8	0.8 × 2 = 1.6	1
2   4	0.6 × 2 = 1.2	1
2   2	0.2 × 2 = 0.4	0
2   1		
1		

repeating

$-273.15_{10} =$ 

sign	exponent	mantissa
1	10000111	100010001001001001

negative  $135_{10}$  15 digits before repeating 8 digits by repetition

b)  $\boxed{2^7 + 2^4 + 2^0}$   $\boxed{2^8 + 2^4 + 2^0}$   
exponent =  $10000111_2 = 135_{10}$   $135 - 127 = 8$  first 9 digits of mantissa  
 $100010001_2 = 273_{10}$   
 $.0010011001001$  is the decimal part  $\boxed{\frac{2^{-3} + 2^{-6} + \dots + 2^{-14}}{2 + 2 + \dots + 2}} = 0.14996 \approx 0.15$   
Therefore, the number is  $-273.15_{10}$

Problem 5.3:

a)

```
remainder :: Int -> Int -> Int
remainder a b = a `mod` b

divide :: Int -> Int -> Int
divide a b = a `div` b

toBase :: Int -> Int -> [Int]
toBase b n
  | divide n b /= 0 = (toBase b (divide n b)) ++ [remainder n b]
  | divide n b == 0 = (n:[])
```

b)

```
fromBase :: Int -> [Int] -> Int
fromBase b n
  | (null n) == False = (fromBase b (tail n)) + ((head n)*(b^((length n)
-1)))
  | (null n) == True = 0
```

c)

```
remainder :: Int -> Int -> Int
remainder a b = a `mod` b

divide :: Int -> Int -> Int
divide a b = a `div` b

showBase :: Int -> Int -> String
showBase b n
  | divide n b /= 0 = (showBase b (divide n b)) ++ (show (remainder n b)
)
  | divide n b == 0 = show n

showBin :: Int -> String
showBin n = showBase 2 n

showOct :: Int -> String
showOct n = showBase 8 n

showHex :: Int -> String
showHex n = showBase 16 n
```

d)

```
readBase :: Int -> String -> Int
readBase b n
  | (null n) == False = (readBase b (tail n)) + (((read n :: Int) `div`
10^((length n)-1)) * (b^((length n)-1)))
  | (null n) == True = 0
```

```
readBin :: String -> Int  
readBin n = readBase 2 n
```

```
readOct :: String -> Int  
readOct n = readBase 8 n
```

```
readHex :: String -> Int  
readHex n = readBase 16 n
```