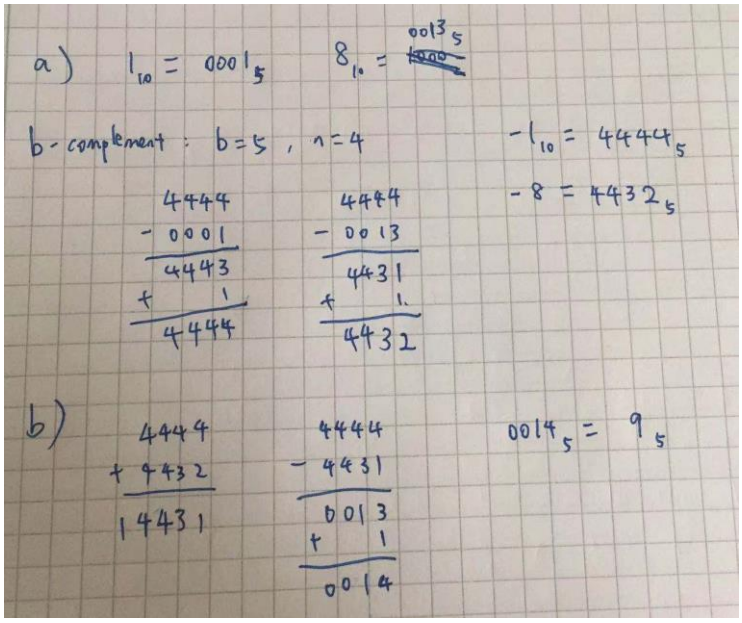
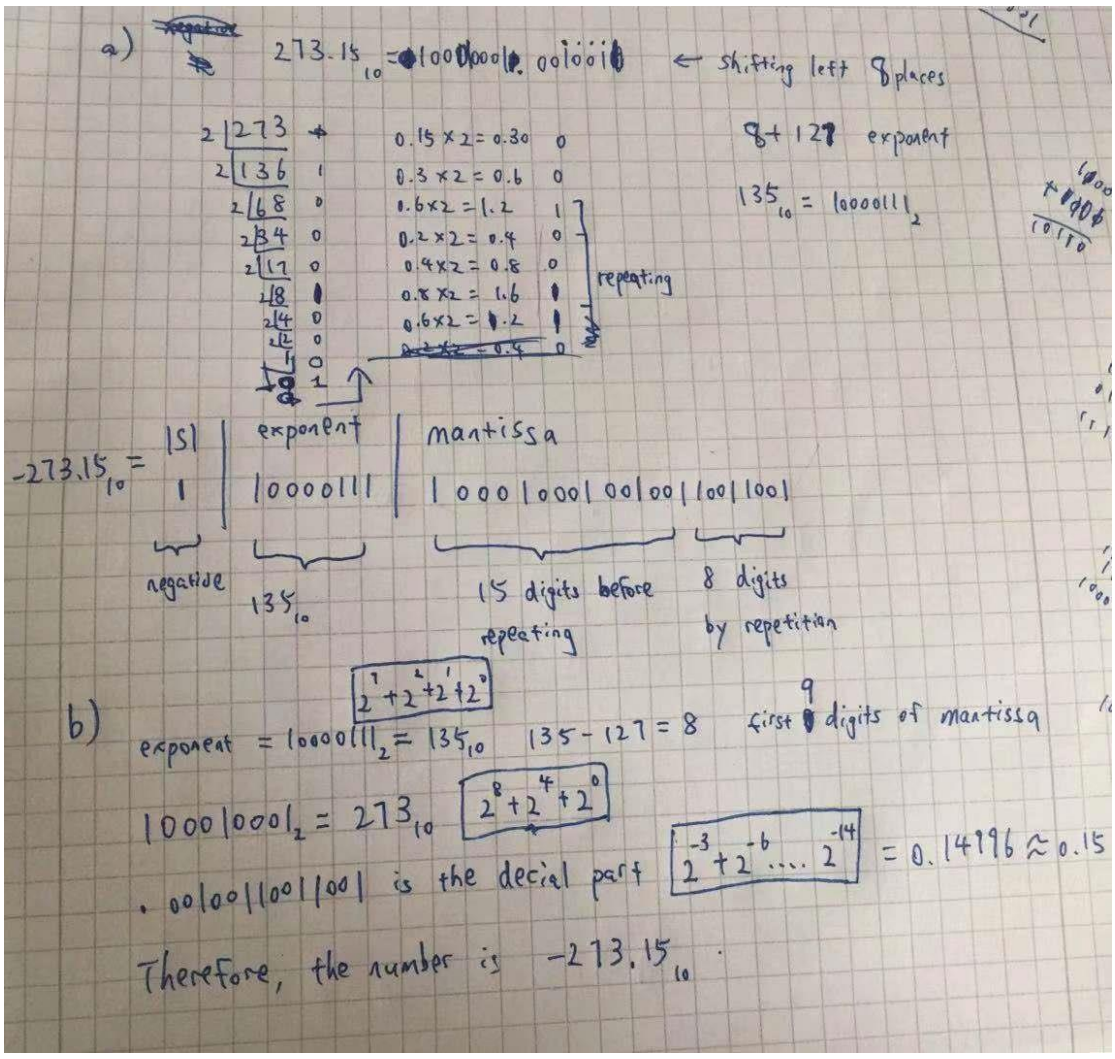
Problem 5.1:



Problem 5.2:



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
```

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

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

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