Chapter Two Exercises:

**Exercise 1) Work through the following examples from this chapter using GHCi**

PS C:\Users\That\_Nein> ghci  
>> GHCi, version 8.2.1: http://www.haskell.org/ghc/ :? for help

>> Prelude> 2+3\*4  
>> 14

>> Prelude> (2+3)\*4  
>> 20

>> Prelude> sqrt(3^2 + 4^2)  
>> 5.0

>> Prelude> head [1,2,3,4,5]  
>> 1

>> Prelude> tail[1,2,3,4,5]  
>> [2,3,4,5]

>> Prelude> [1,2,3,4,5] !! 2  
>> 3

>> Prelude> take 3[1,2,3,4,5]  
>> [1,2,3]

>> Prelude> drop 3[1,2,3,4,5]  
>> [4,5]

>> Prelude> length[1,2,3,4,5]  
>> 5

>> Prelude> sum[1,2,3,4,5]  
>> 15

>> Prelude> product[1,2,3,4,5]  
>> 120

>> Prelude> [1,2,3]++[4,5]  
>> [1,2,3,4,5]

>> Prelude> reverse[1,2,3,4,5]  
>> [5,4,3,2,1]

PS C:\Users\That\_Nein\desktop> ghci test.hs  
GHCi, version 8.2.1: http://www.haskell.org/ghc/ :? for help  
[1 of 1] Compiling Main ( test.hs, interpreted )  
Ok, 1 module loaded.

\*Main> quadruple 10  
40

\*Main> take (double 2) [1,2,3,4,5]  
[1,2,3,4]

\*Main> :reload  
[1 of 1] Compiling Main ( test.hs, interpreted )  
test.hs:5:21: error:  
 \* Syntax error on 'div'  
 Perhaps you intended to use TemplateHaskell or TemplateHaskellQuotes  
 \* In the Template Haskell quotation 'div'  
 |  
5 | average ns = sum ns 'div' length ns  
 | ^^^^^  
Failed, 0 modules loaded.

Prelude> :reload  
[1 of 1] Compiling Main ( test.hs, interpreted )  
Ok, 1 module loaded.

\*Main> factorial 10  
3628800

\*Main> average [1,2,3,4,5]  
3

------------------------------------------------------------------------------------------------

**Exercise 2) Parenthesis the following numeric expressions**

PS C:\Users\That\_Nein\desktop> ghci  
GHCi, version 8.2.1: http://www.haskell.org/ghc/ :? for help

Prelude> 2^3\*4  
32  
Prelude> (2^3)\*4  
32

Prelude> 2\*3+4\*5  
26  
Prelude> (2\*3)+(4\*5)  
26

Prelude> 2+3\*4^5  
3074  
Prelude> 2+(3\*(4^5))  
3074

**Exercise 3) Fix the syntax errors and ensure the function works.**

n = a `div` length xs

where

a = 10

xs = [1,2,3,4,5]

**Errors found: n must be lowercase, must use ` for quotes instead of ‘, the lines under where must start at the same position.**

PS C:\Users\That\_Nein\desktop> ghci ex2.hs  
GHCi, version 8.2.1: http://www.haskell.org/ghc/ :? for help  
[1 of 1] Compiling Main ( ex2.hs, interpreted )  
Ok, 1 module loaded.

\*Main> n  
2

**Exercise 4) The library function last selects the last element of a non-empty list; for example, last [1,2,3,4,5] = 5. Show how the function last should be defined in terms of the other library functions introduced in this chapter. Two definitions of last:**

lastOne :: [a] -> a  
lastOne x = head(reverse x)

lastTwo :: [a] -> a  
lastTwo x = x !! ( length x - 1)

PS C:\Users\That\_Nein\desktop> ghci ex4.hs  
GHCi, version 8.2.1: http://www.haskell.org/ghc/ :? for help  
[1 of 1] Compiling Main ( ex4.hs, interpreted )  
Ok, 1 module loaded.

\*Main> lastOne[1,2,3,4,5]  
5

\*Main> lastTwo[1,2,3,4,5]  
5

**Exercise 5) The library function init removes the last element from a non-empty list; for example, init [1,2,3,4,5] = [1,2,3,4]. Show how init could similarly be defined in two different ways.**

initOne x = reverse(tail(reverse x))

initTwo x = reverse(drop 1(reverse x))

\*Main> initOne[1,2,3,4,5]  
[1,2,3,4]

\*Main> :reload  
[1 of 1] Compiling Main ( ex5.hs, interpreted )  
Ok, 1 module loaded.  
\*Main> initTwo[1,2,3,4,5]  
[1,2,3,4]

**Chapter Three Exercises:**

**Exercise 3) What are the types of the following functions?**  
  
**second xs = head (tail xs)**

second :: [a] -> a

**swap (x,y) = (y,x)**

swap :: (a,b) -> (b,a)

**pair x y = (x,y)**

pair :: a -> b -> (a,b)

**double x = x \* 2**

double :: Num a => a -> a

**palindrome xs = reverse xs == xs**

palindrome :: Eq a => [a] -> Bool

**twice f x = f (f x)**

twice: ((a -> a) -> a) -> a

**Exercise 4) Check your answers to the previous three problems using GHC:**

**Ch3.hs**bools :: [Bool]  
bools = [True]

nums :: [[Int]]  
nums = [[42]]

add :: Int -> Int -> Int -> Int  
add a b c = a \* b + c

copy :: a -> (a,a)  
copy a = (a,a)

apply :: (a -> b) -> a -> b  
apply a b = a b

second :: [a] -> a  
second xs = head (tail xs)

swap :: (a,b) -> (b,a)  
swap (x,y) = (y,x)

pair :: a -> b -> (a,b)  
pair x y = (x,y)

double :: Num a => a -> a  
double x = x \* 2

palindrome :: Eq a => [a] -> Bool  
palindrome xs = reverse xs == xs

twice :: (a -> a) -> a -> a  
twice f x = f (f x)

**Exercise1)**

Prelude> :t ['a', 'b', 'c']  
['a', 'b', 'c'] :: [Char]

Prelude> :t ('a', 'b', 'c')  
('a', 'b', 'c') :: (Char, Char, Char)

Prelude> :t [(False, '0'),(True, '1')]  
[(False, '0'),(True, '1')] :: [(Bool, Char)]

Prelude> :t ([False, True],['0', '1'])  
([False, True],['0', '1']) :: ([Bool], [Char])

Prelude> :t [tail, init, reverse]  
[tail, init, reverse] :: [[a] -> [a]]

**Exercise2)**

PS C:\Users\That\_Nein\desktop> ghci ch3.hs  
GHCi, version 8.2.1: http://www.haskell.org/ghc/ :? for help  
[1 of 1] Compiling Main ( ch3.hs, interpreted )  
Ok, 1 module loaded.

\*Main> :t bools  
bools :: [Bool]

\*Main> :t nums  
nums :: [[Int]]

\*Main> :t add  
add :: Int -> Int -> Int -> Int

\*Main> :t copy  
copy :: a -> (a, a)

\*Main> :t apply  
apply :: (a -> b) -> a -> b

**Exercise 3)**

\*Main> :t second  
second :: [a] -> a

\*Main> :t swap  
swap :: (a, b) -> (b, a)

\*Main> :t pair  
pair :: a -> b -> (a, b)

\*Main> :t double  
double :: Num a => a -> a

\*Main> :t palindrome  
palindrome :: Eq a => [a] -> Bool

\*Main> :reload  
[1 of 1] Compiling Main ( ch3.hs, interpreted )  
Ok, 1 module loaded.  
\*Main> :t twice  
twice :: (a -> a) -> a -> a