Exercises and Solutions
Types and Classes

Exercise 1:
What are the types of the following values?

```
['a', 'b', 'c']
('a', 'b', 'c')
[(False, '0'), (True, '1')]
(['1', '0'], ['0', '1'])
[tail, init, reverse]
```

Use GHCi (:t) to check your solutions.

Solution 1:

```
['a', 'b', 'c'] :: [Char]
('a', 'b', 'c') :: (Char, Char, Char)
[(False, '0'), (True, '1')] :: [(Bool, Char)]
(['1', '0'], ['0', '1']) :: ([String], [String])
[tail, init, reverse] :: [[a] -> [a]]
```

Exercise 2:
Write down definitions that have the following types:

```
bools :: [Bool]
nums :: [[Int]]
add :: Int -> Int -> Int -> Int
copy :: a -> (a, a)
apply :: (a -> b) -> a -> b
```

Check your solutions using GHCi.

Solution 2:

```
bools :: [Bool]
bools = [True, False]

nums :: [[Int]]
nums = [[1,2,3], [2,3,4]]

add :: Int -> Int -> Int -> Int
add x y z = x + y + z
```

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

apply :: (a -> b) -> a -> b
apply f x = f x
```

Exercise 3:
What are the types of the following functions?

```
copy x = (x, x)
second xs = head (tail xs)
swap (x,y) = (y,x)
pair x y = (x,y)
double x = x * 2
palindrome xs = reverse xs == xs
twice f x = f (f x)
```

Use GHCi (‘:t‘) to check their types.

Solution 3:

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

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)
```

Exercise 4:

When writing the following functions, what type class should be used in the class constraint?

```
(i) multiply :: ___ => a -> a -> a
(ii) areEqual :: ___ => a -> a -> Bool
(iii) sort :: ___ => [a] -> [a]
(iv) maxList :: ___ => [a] -> a
```

Solution 4:

```
(i) multiply :: Num a => a -> a -> a
(ii) areEqual :: Eq a => a -> a -> Bool
(iii) sort :: Ord a => [a] -> [a]
(iv) maxList :: Ord a => [a] -> a
```