Exercises
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 answers.

Exercise 2

Write down definitions that have the following types. It does not matter that the definitions actually do as long as they are type correct:

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

Check your answers using GHCi. You can do this using a script or by using the let construct:

```
GHCi, version 8.2.1: http://www.haskell.org/ghc/  :? for help
Prelude> let bools = [True,False] in bools :: [Bool]
[True,False]
Prelude>
```

Exercise 3

What are the types of the following functions?

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

The easiest way to check this is to use the *:t* at the console. You can also check this by putting these in a script with the type. If they are not consistent, you will get an error when you run/load the script.
Also, take care to include the necessary class constraints (e.g. Eq a => when you are testing for equality)if the functions are defined using overloaded operators.