```
module Workshop2 where

import Data.List

infix 1 ==>

(==>) :: Bool -> Bool -> Bool
p ==> q = (not p) || q

forall = flip all
```

Workshop Testing and Formal Methods, Week 2

This workshop is about understanding fundamental concepts in algorithm specification and algorithm design.

The focus is on pre- and postcondition specifications.

The first exercise uses a sudoku example that you will encounter further on in the course.

Question 1 A sudoku is a 9×9 matrix of numbers in $\{1, \ldots, 9\}$, possibly including blanks, satisfying certain constraints. A *sudoku problem* is a sudoku containing blanks, but otherwise satisfying the sudoku constraints. The sudoku solver transforms the problem into a solution.

Give a Hoare triple for a sudoku solver. If the solver is called P, the Hoare triple consists of

$$\{\text{precondition}\}$$
$$P$$
$$\{\text{postcondition}\}$$

The precondition of the sudoku solver is that the input is a correct sudoku problem.

The postcondition of the sudoku solver is that the transformed input is a solution to the initial problem.

State the pre- and postconditions as clearly and formally as possible.

Question 2 A function of type $a \to a$ (a unary function with arguments and values of the same type) can be tested with test properties of the type $a \to \text{Bool}$ or of the type $a \to a \to \text{Bool}$.

We will consider test properties of type $a \to \text{Bool}$.

Define the following predicate on test properties:

```
stronger, weaker :: [a] -> (a -> Bool) -> (a -> Bool) -> Bool
stronger xs p q = forall xs (\ x -> p x ==> q x)
weaker   xs p q = stronger xs q p
```

This gives:

```
*Workshop2> stronger [1..10] odd odd
True
*Workshop2> stronger [1..10] odd (\ x -> odd x || x > 5)
True
*Workshop2> stronger [1..10] odd (\ x -> odd x && x > 5)
False
*Workshop2> weaker [1..10] odd odd
True
*Workshop2> weaker [1..10] odd (\ x -> odd x || x > 5)
False
*Workshop2> weaker [1..10] odd (\ x -> odd x && x > 5)
True
```

Predict the output of the following Haskell tests:

```
test1 = stronger [1..10] (\ x -> even x && x > 3) even
test2 = stronger [1..10] (\ x -> even x || x > 3) even
test3 = stronger [1..10] (\ x -> (even x && x > 3) || even x) even
test4 = stronger [1..10] even (\ x -> (even x && x > 3) || even x)
```

Question 3 Now suppose $\{p\}\ f\ \{q\}$ holds for some function $f : a \to a$ and a pair of properties p and q.

Recall the meaning of $\{p\}\ f\ \{q\}$:

For every possible argument x for f it is the case that if x has property p then $f(x)$ has property q.

1. If p' is stronger that p, does it follow that $\{p'\}\ f\ \{q\}$ still holds?

2. If p' is weaker that p, does it follow that $\{p'\}\ f\ \{q\}$ still holds?

3. If q' is stronger that q, does it follow that $\{p\}\ f\ \{q'\}$ still holds?

4. If q' is weaker that q, does it follow that $\{p\}\ f\ \{q'\}$ still holds?

Question 4 Which of the following properties is stronger?

1. $\lambda x \mapsto x = 0$ and $\lambda x \mapsto x \geq 0$

2. $\lambda x \mapsto x \neq 0$ and $\lambda x \mapsto x > 3$

3. $\lambda x \mapsto x \neq 0$ and $\lambda x \mapsto x < 3$

4. $\lambda x \mapsto x^3 + 7x^2 \geq 3$ and $\lambda x \mapsto \bot$

5. $\lambda x \mapsto x \geq 2 \vee x \leq 3$ and $\lambda x \mapsto x \geq 2$

6. $\lambda x \mapsto x \geq 2 \wedge x \leq 3$ and $\lambda x \mapsto x \geq 2$

Question 5 Implement all properties from the previous question as Haskell functions of type `Int -> Bool`. Note: this is a pen and paper exercise: just write out the definitions. If you have a computer, this allows you to check your answers to the previous exercise, on some small domain like $[(-10)..10]$.

Question 6 Now that we know what weaker and stronger means, we can talk about the weakest property p for which

$$\{p\}\ f\ \{q\}$$

holds, for a given function f and a given postcondition property q.

Example: the weakest p for which

$$\{p\}\lambda x \mapsto 2 * x + 4\ \ \{\lambda x \mapsto 0 \leq x < 8\}$$

holds is $\lambda x \mapsto -2 \leq x < 2$.

Note: $\lambda x \mapsto 0 \leq x < 8$ has to hold. The recipe for finding out when that is the case is as follows.

Use the function $\lambda x \mapsto 2 * x + 4$ as a *substitution*: substitute the right-hand side $2 * x + 4$ for x in the postcondition q to get the weakest precondition, and simplify.

Work out the weakest preconditions for the following triples.

1. $\{\cdots\}\ \lambda x \mapsto x{+}1\ \{\,\lambda x \mapsto 2x - 1 = A\,\}$

2. $\{\cdots\}\ \lambda x \mapsto x * x + 1\ \{\,\lambda x \mapsto x = 10\,\}$

3. $\{\cdots\}\ \lambda x \mapsto x{+}y\ \{\,\lambda x \mapsto x{-}y = 7\,\}$
4. $\{\cdots\}\ \lambda x \mapsto x{+}y\ \{\,\lambda x \mapsto x \geq y\,\}$
5. $\{\cdots\}\ \lambda x \mapsto -x\ \{\,\lambda x \mapsto x \geq 0\,\}$

Question 7 Show the following:

1. $\{\,\lambda n \mapsto x = n^2\,\}\ \lambda n \mapsto n{+}1\ \{\,\lambda n \mapsto x = (n{-}1)^2\,\}$
2. $\{\,\lambda x \mapsto A = x\,\}\ \lambda x \mapsto x{+}1\ \{\,\lambda x \mapsto A = x{-}1\,\}$
3. $\{\,\lambda x \mapsto x \geq 0\,\}\ \lambda x \mapsto x{+}y\ \{\,\lambda x \mapsto x \geq y\,\}$
4. $\{\,\lambda x \mapsto 0 \leq x < 100\,\}\ \lambda x \mapsto x{+}1\ \{\,\lambda x \mapsto 0 \leq x \leq 100\,\}$
5. $\{\,\lambda n \mapsto x = (n{+}1)^2 \wedge n = A\,\}\ \lambda n \mapsto n{+}1\ \{\,\lambda n \mapsto x = n^2 \wedge n = A{+}1\,\}$