# Theory of Programming: Homework #2

Due date: 2024/01/10

## Written part

This portion requires written proofs (in Chinese or English, not on Isabelle, even if they refer to problems in the book *Concrete Semantics*).

1. (see Exercise 12.5) Consider the following program with given pre- and postcondition:

$$
\begin{array}{l}
\{x = X \wedge y = Y \wedge 0 \le X\} \\
\textbf{while } 0 < x \textbf{ do} \\
\quad x := x - 1; \\
\quad y := y - 1 \\
\{y = Y - X\}
\end{array}
$$

   Find an invariant of the while loop for proving this Hoare triple. Write the verification conditions that would be generated and prove them informally.

2. (see Exercise 12.7) Do the same as in Exercise 1 for the following program:

$$
\begin{array}{l}
\{x = X \wedge 0 \le X\} \\
r := 0; \\
r_2 := 1; \\
\textbf{while } \neg(x < r_2) \textbf{ do} \\
\quad r := r + 1; \\
\quad r_2 := r_2 + (r + r + 1) \\
\{x = X \wedge r^2 \le X \wedge X < (r + 1)^2\}
\end{array}
$$

## Isabelle part

For this part, submit Isabelle `.thy` files on the class website.

1. (Exercise 2.3) Define a function $count :: 'a \Rightarrow 'a\ list \Rightarrow nat$ that counts the number of occurrences of an element in a list. Prove $count\ x\ xs \leq length\ xs$.

2. (Exercise 2.7) Define a new type $'a\ tree2$ of binary trees where values are also stored in the leaves of the tree. Also reformulate the mirror function accordingly (see page 16 for the original definition of the mirror function). Define two functions $pre\_order$ and $post\_order$ of type $'a\ tree2 \Rightarrow 'a\ list$ that traverse a tree and collect all stored values in the respective order in a list. Prove $pre\_order\ (mirror\ t) = rev\ (post\_order\ t)$.

   Note: preorder traversal means first visiting the root, then recursively visiting the left and right subtree. Postorder traversal means first recursively visiting the left and right subtree (in that order), then visit the root.