Tutorial 2

Exercise 1

(Kingston, exercise 1.1, page 11).

g$(\mathbf{n})\{$
if $n \leq 1$ **then**
  $Result := n$
else
  $Result := 5 \times g(n-1) - 6 \times g(n-2)$
end if
$\}$

Claim: For all integers $n \geq 0$ it holds that **g**$(n) = 3^n - 2^n$ (i.e. for all n greater than or equal to 0 the function **g(**n**)** returns $3^n - 2^n$).

Proof: The proof is by induction on n.

Basic step: Let $n = 0$. Then $g(0)$ returns 0 and $3^0 - 2^0 = 0$ $\surd$.
Let $n = 1$. Then $g(1)$ returns 1 and $3^1 - 2^1 = 1$ $\surd$.

Inductive step: Let $n > 1$. IH="$g(j)$ returns $3^j - 2^j$ for all integers j in the range $0 \leq j \leq n-1$", which is in accordance with the second principle of mathematical induction (see Rosen p. 196 available in the course folder). From the induction hypothesis (IH), we can assume that $g(n-1)$ returns $3^{n-1} - 2^{n-1}$ and that $g(n-2)$ returns $3^{n-2} - 2^{n-2}$. Now we show that the value returned by the function $g(n)$ equals $3^n - 2^n$ under this assumption. Because $n > 1$ the test of the `IF` statement is false, thus $g(n)$ returns $5 \cdot g(n-1) - 6 \cdot g(n-2)$, and

$$\begin{aligned}
5 \cdot g(n-1) - 6 \cdot g(n-2) &\overset{\text{IH}}{=} 5 \cdot (3^{n-1} - 2^{n-1}) - 6 \cdot (3^{n-2} - 2^{n-2}) \\
&= 5 \cdot (3 \cdot 3^{n-2} - 2 \cdot 2^{n-2}) - 6 \cdot (3^{n-2} - 2^{n-2}) \\
&= 15 \cdot 3^{n-2} - 10 \cdot 2^{n-2} - 6 \cdot 3^{n-2} + 6 \cdot 2^{n-2} \\
&= 9 \cdot 3^{n-2} - 4 \cdot 2^{n-2} \\
&= 3^2 \cdot 3^{n-2} - 2^2 \cdot 2^{n-2} \\
&= 3^n - 2^n \surd
\end{aligned}$$

and this is indeed equal to $3^n - 2^n$ as required.

Exercise 2

Pre: $X = a, Y = b$.

Post: $X = b, Y = a$.

Proof: The proof is performed by the assertion method:

$\{X = a, Y = b\}$

- $X := X + Y$

$\{X = a + b, Y = b\}$

- $Y := X - Y$

$\{X = a + b, Y = a\}$

- $X := X - Y$

$\{X = b, Y = a\}$

Exercise 3

- No, the algorithm is not totally correct, e.g. -2 satisfies the pre-condition, however the algorithm does not terminate on this input.
- Yes. Let n be an arbitrary integer. If $n < 0$ then the algorithm does not terminate and hence no post-condition has to be checked. If $n \geq 0$ then the algorithm terminates and outputs $n!$ which satisfies the post-condition.

Exercise 4

Claim: The specification:

pre: $a \leq b + 1$

post: $entries.item(a) \leq entries.item(a+1) \leq \ldots \leq entries.item(b)$

is satisfied by the algorithm of exercise 1.2 on page 11.

Proof: The proof is by induction on n (the length of the array, i.e. $n = b - a + 1$). We will prove the claim for a stronger post-condition post' (this will of course imply that also post is satisfied):

post': "$selection\_sort(a, b)$ changes only the values between a and b and the changed values are some permutation of the original ones such that $entries.item(a) \leq entries.item(a+1) \leq \ldots \leq entries.item(b)$"

Basic step: Let $n = 0$. This implies that $a = b + 1$ (the array is empty and hence sorted) and the algorithm does nothing as required $\surd$.

Inductive step: Let $n \geq 1$.

IH="For all j, $0 \leq j \leq n-1$, and for all a and b such that $j = b - a + 1$ it holds that $selection\_sort(a, b)$ changes only the values between a and b and the changed values are some permutation of the original ones such that $entries.item(a) \leq entries.item(a+1) \leq \ldots \leq entries.item(b)$"

Let us now consider a call of $selection\_sort(a, b)$ such such that $b-a+1 = n$. We want to show that after its execution the condition post' will be true.

From IH we can assume that $selection\_sort(a+1, b)$ sorts the entries between $a+1$ and b ($b-(a+1)+1 < b-a+1$) such that $entries.item(a+1) \leq entries.item(a+2) \leq \ldots \leq entries.item(b)$ and nothing else is changed.

In the call of $selection\_sort(a, b)$ the else branch of the if-command is executed ($n \geq 1$) and the entries $a+1$ to b are sorted by the recursive call $selection\_sort(a+1, b)$ (hence by the IH: $entries.item(a+1) \leq entries.item(a+2) \leq \ldots \leq entries.item(b)$) and we also know that $entries.item(a) \leq entires.item(i)$ for all i, $a \leq i \leq b$ (this holds after performing min_index and swap), hence in particular also $entries.item(a) \leq entries.item(a+1)$. This means that post' is satisfied $\surd$.