Question 1 (6 marks).

[Learning Goal: Prove that a specification of a program using arrays is satisfied under partial correctness.]

Consider the following specification:

\[
A[1] = t ; \\
\]

Prove that this specification is satisfied under partial correctness. Show all the needed details of your program annotation, and explicitly prove any implications required by your annotation.
Question 2 (15 marks).

[Learning Goal: Prove that a specification of a program using arrays is satisfied under total correctness.]

The code below might be used as part of an “bubble sort”. It scans an array \( A \) and exchanges some of the out-of-order elements; in particular, it moves the largest element in the array to the end (position \( n \)).

\[
\{ \text{Perm}(A, A_0) \land n > 0 \} \\
i = 1 \\
\text{while } (i < n) \\
\{ \\
\text{if } (A[i] > A[i+1]) \\
\quad t = A[i] \\
\quad A[i] = A[i+1] \\
\quad A[i+1] = t \\
\} \\
i = i+1; \\
\} \\
\{ \text{Perm}(A, A_0) \land \text{Max}(A, n) \} 
\]

The relations in the pre- and post-condition are

- \( \text{Perm}(X, Y) \): Arrays \( X \) and \( Y \) are permutations of one another.
- \( \text{Max}(X, m) \): In array \( X \), the largest of the first \( m \) elements is at position \( m \).

To work with these, use the formula

\[
\forall j((1 \leq j \land j < m) \rightarrow (X[j] \leq X[m]))
\]

as the definition of \( \text{Max}(X, m) \). Use the following axioms for \( \text{Perm} \): for every array \( B, C \) and \( D \), and integers \( x \) and \( y \),

- P1: \( \text{Perm}(B, B) \)
- P2: \( \text{Perm}(B, C) \rightarrow \text{Perm}(C, B) \).
- P3: \( (\text{Perm}(B, C) \land \text{Perm}(C, D)) \rightarrow \text{Perm}(B, D) \).
- P4: \( \text{Perm}(B, B\{x \leftarrow B[y]\}{y \leftarrow B[x]}) \).

For the first part of the question (on the next page), fill in the blanks with suitable annotations. The loop invariant has been filled in for you.
Q2(a)

\[ n > 0 \land \text{Perm}(A, A_0) \]  
\[ i = 1 ; \]

\[ \text{Perm}(A, A_0) \land (i \leq n \land \text{Max}(A, i)) \]

while ( i < n ) {

\[ t = A[i] ; \]

\[ A[i] = A[i+1] ; \]

\[ A[i+1] = t ; \]

\[ i = i + 1 ; \]

\[ \text{Perm}(A, A_0) \land \forall j((1 \leq j \land j < n) \rightarrow (A[j] \leq A[n])) \]
Q2 (b) Justify each of the “implied” conditions from your template.
Formal proofs are NOT required; the markers will assume that you know basic arithmetic.
Do, however, mention the key points.
Q2 (c) Show that the program is totally correct.
You may assume the results for the previous parts of this question, whether you solved them or not.
Question 3 (10 marks).

[Learning Goal: Prove that a decision problem is decidable.]

Let $\mathbb{N}$ denote the set of Natural numbers, that is $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$. Let $S_1, S_2 \subseteq \mathbb{N}$ be sets such that membership in each of $S_1$ and $S_2$ is decidable. Prove that membership in each of the following sets is also decidable:

(a) $S_1 \cap S_2 = \{n \in \mathbb{N} \mid n \in S_1 \text{ and } n \in S_2\}$, the intersection of $S_1$ and $S_2$ in $\mathbb{N}$. 

[3]
\(\mathbb{N} \setminus (S_1 \cup S_2) = \{n \in \mathbb{N} \mid n \notin (S_1 \cup S_2)\}\), the complement of \((S_1 \cup S_2)\) in \(\mathbb{N}\).
\[ \{ x + y \mid x \in S_1 \text{ and } y \in S_2 \} \]
Question 4 (6 marks).
[Learning Goal: Prove that a decision problem is undecidable.]

Assume that every program in this question consumes a single Natural number as its input. Assume further that if a program halts when run with a particular input, then it follows that the program returns a single Natural number as its output.

Definition: Let $P_1$ and $P_2$ be programs. Then we say that $P_1$ and $P_2$ agree on all inputs if, for every input $x$, it follows that either

- $P_1$ and $P_2$ both run forever when processing input $x$, or
- $P_1$ and $P_2$ both halt when processing input $x$, and $P_1$ and $P_2$ return the same output when processing input $x$.

Consider the decision problem

given any two programs $P_1$ and $P_2$, do $P_1$ and $P_2$ agree on all inputs?

Prove that this decision problem is undecidable.

Hint: For a contradiction, suppose that the given decision problem is decidable. Then show that with this assumption, you can decide the Halting Problem.