Question 1 Prove that assigning Volpano’s information flow type of class 13 to a program is undecidable (and so any type system like Volpano’s information flow type system will reject infinitely many programs with no security leak but which cannot be proved to be typable by the type system).

Question 2 Provide an example of a very simple program which is secure but

- fails to be proved secure by the Dennis flow analysis of class 12;
- fails to be typed secure by the Volpano-Irvine-Smith type system of class 13;
- is provably secure by the security static analysis of class 14.

Question 3 Consider the following erroneous rule for constants (L is low, the infimum of the security lattice) for the static program flow analysis of class 12.

\[
\begin{align*}
\{1:\} & \quad a := \text{cte} \quad \{2:\} \quad PC_2 = PC_1 \\
& \quad I_2(a) = L
\end{align*}
\]

Provide a counter-example to this rule.

Question 4 Consider the rule for conditional in the static program flow analysis of class 12.

\[
\begin{align*}
\{1:\} & \quad \text{if } B(o_1, \ldots, o_n) \text{ then} \\
& \quad \{2:\} \quad S \quad \{3:\} \quad PC_2 = PC_1 \oplus I_1(o_i) \oplus \ldots \oplus I_1(o_n) \\
& \quad \quad I_2(o_i) = PC_2, \quad i = 1, \ldots, n \\
\text{else} & \quad \{4:\} \quad S' \quad \{5:\} \quad PC_4 = PC_1 \oplus I_1(o_i) \oplus \ldots \oplus I_1(o_n) \\
& \quad \quad I_4(o_i) = PC_4, \quad i = 1, \ldots, n \\
& \quad \text{fi} \\
\{6:\} & \quad PC_6 = PC_1 \\
& \quad I_6(o_i) = I_3(o_i) \oplus I_5(o_i), \quad i = 1, \ldots, n
\end{align*}
\]
Explain why the information leak, if any, in the test $B(o_1, \ldots, o_n)$ can propagate beyond program $\{6:\}$ despite the fact that $PC_6 = PC_1$ stating that the information leaked after the test is the same as that leaked before the test. □

**Question 5** Consider again the static program flow analysis of class 12. Propose a rule for the local variable declaration which scope/visibility is the following block of declaration.

$$\{1:\} \ \text{begin} \ \var x := f(o_1, \ldots, o_n); \ \{2:\} \ S \ \{3:\} \ \text{end}; \ \{4:\}$$

Note that if one of the variables $o_1, \ldots, o_n$ is $x$ in the initialization of the declaration $\var x := f(o_1, \ldots, o_n)$; then it cannot be the new fresh $x$ local to the block but it must be one of the global variables $o_1, \ldots, o_n$ visible at program point $\{1:\}$. However in statement $S$, any instance of $x$ is that of the local variable (hiding the global one amongst the $o_1, \ldots, o_n$, if any, which is not visible in the block beyond point $\{2:\}$ after the declaration of the fresh variable $x$). □