1.) An undirected graph $(V, E)$ is called a mirror graph if the following conditions hold:

- $V$ can be partitioned into two sets of equal size $V' = \{v'_1, \ldots , v'_n\}$ and $V'' = \{v''_1, \ldots , v''_n\}$;
- for all $i \in \{1, \ldots , n\}$ the edge $[v'_i, v''_i]$ is in $E$;
- for all $i, j \in \{1, \ldots , n\}$, $[v'_i, v'_j] \in E$ if and only if $[v''_i, v''_j] \in E$;
- no other edges are contained in $E$.

In words, a mirror graph has each vertex from $V'$ connected to a clone from $V''$ and the graph over $V'$ is mirrored in $V''$.

Consider the following new problem $3\text{COLMG}$ and recall the NP-complete problem $3\text{COL}$ defined below:

### 3-COLORABILITY-MIRROR GRAPH ($3\text{COLMG}$)

**INSTANCE:** A mirror-graph $G = (V, E)$.

**QUESTION:** Does there exists a function $\mu$ from vertices in $V$ to values in $\{0, 1, 2\}$ such that $\mu(v_1) \neq \mu(v_2)$ for any edge $[v_1, v_2] \in E$.

### 3-COLORABILITY ($3\text{COL}$)

**INSTANCE:** An undirected graph $G = (V, E)$.

**QUESTION:** Does there exists a function $\mu$ from vertices in $V$ to values in $\{0, 1, 2\}$ such that $\mu(v_1) \neq \mu(v_2)$ for any edge $[v_1, v_2] \in E$.

(a) The following function $f$ provides a polynomial-time many-one reduction from $3\text{COL}$ to $3\text{COLMG}$: for a directed graph $G = (\{v_1, \ldots , v_n\}, E)$, let

\[
    f(G) = (\{v'_1, v''_1, \ldots , v'_n, v''_n\},
            \{[v'_i, v''_i] | i = 1 \ldots n\} \cup
            \{[v'_i, v'_j], [v''_i, v''_j] | [v_i, v_j] \in E\})
\]

Show that $G$ is a yes-instance of $3\text{COL} \iff f(G)$ is a yes-instance of $3\text{COLMG}$.

(9 points)

(b) Tick the correct statements (for ticking a correct statement a certain number of points is given; ticking an incorrect statement results in a substraction of the same amount; you cannot go below 0 points):

- Given that $3\text{COL}$ is in NP, our reduction shows that $3\text{COLMG}$ is also in NP.
- Given that $3\text{COL}$ is in NP, our reduction shows that $3\text{COLMG}$ is NP-hard.
- Given that $3\text{COL}$ is NP-hard, our reduction shows that $3\text{COLMG}$ is also NP-hard.
- Given that $3\text{COL}$ is NP-hard, our reduction shows that $3\text{COLMG}$ is in NP.
- Since $3\text{COLMG}$ is a special case of $3\text{COL}$, $3\text{COLMG}$ must be contained in NP.
- Since $3\text{COLMG}$ is a special case of $3\text{COL}$, $3\text{COLMG}$ must be NP-hard.

(6 points)
2.) (a) First define the concept of a theory and of a $\mathcal{T}$-interpretation. Then use them to define:

i. the $\mathcal{T}$-satisfiability of a formula;
ii. the $\mathcal{T}$-validity of a formula.

Additionally define the completeness of a theory $\mathcal{T}$.  

(3 points)

(b) Consider the function $M$, defined as follows.

**Algorithm 1: The function $M$**

<table>
<thead>
<tr>
<th>Input: $x, y$, two positive integers</th>
<th>Output: The computed positive integer value for $x, y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 if $x == 1$ then</td>
<td>2 return $2y$;</td>
</tr>
<tr>
<td>3 else if $y == 1$ then</td>
<td>4 return $x$;</td>
</tr>
<tr>
<td>5 else return $M(x - 1, M(x, y - 1))$;</td>
<td></td>
</tr>
</tbody>
</table>

Let $\mathbb{N}$ denote the natural numbers without 0. Use well-founded induction to show

$$\forall x, y \ ( (x \in \mathbb{N} \land y \in \mathbb{N}) \rightarrow M(x, y) \geq 2y).$$

(10 points)

(c) Suppose $M_c$ is a correct implementation of $M$ in the C programming language with $x$ and $y$ of type unsigned integers of size 32 bit (i.e., of type `uint32_t`). Is

$$M(x', y') = M_c(x', y')$$

true for all integers $x', y'$ satisfying $1 \leq x', y' \leq \text{UINT32}_\text{MAX}$, where `UINT32_MAX` is the largest value for a variable of type `uint32_t`?

If so, then prove this fact. Otherwise provide a counterexample with an exact explanation of what is computed and what is happening.  

(2 points)
3.) (a) Let $p$ be the following program:

\[
\begin{align*}
y &:= n; \\
\textbf{while} & \quad y > 0 \textbf{ do} \\
x &:= x - 4 \times y + 2; \\
y &:= y - 1 \\
\textbf{od}
\end{align*}
\]

Give a loop invariant for the \textbf{while} loop in $p$ and prove the validity of the partial correctness triple $\{ n > 0 \land x = 2 \times n^2 + 1 \} \ p \ {x = 1}$.  

(9 points)

(b) Provide a non-trivial pre-condition $A$ and a non-trivial post-condition $B$, such that the total correctness triple $\{A\} \ p \ {B}$ is valid. Trivial means equivalent to \textbf{true} or \textbf{false}, so your pre-condition $A$ and postcondition $B$ should not be equivalent to \textbf{true} or \textbf{false}. The program $p$ is given below.

Program $p$:

\[
\begin{align*}
y &:= x + y; \\
\textbf{while} & \quad x \neq y \textbf{ do} \\
y &:= y + 1; \\
x &:= x + 3; \\
\textbf{od}
\end{align*}
\]

(2 points)

(c) Provide a state $\sigma$ such that $\sigma \models [N \geq 0] \textbf{abort} \ [B]$. In case such a $\sigma$ does not exist, explain why there exists no such $\sigma$.  

(2 points)

(d) What went wrong in the following argumentation:

"In order to prove $\{A\} \ p \ {B}$ for any program $p$, we can simply start at the bottom of $p$ and compute the weakest liberal precondition of the last expression with respect to $B$. We use the result as the new postcondition and work my way up to the first line of code. If the weakest liberal precondition of the first line is implied by $A$, we successfully proved partial correctness of $\{A\} \ p \ {B}$."

Give a concise answer, 2-3 sentences suffice! 

(2 points)
4.) (a) Provide a non-empty simulation relation \( H \) that witnesses \( M_1 \leq M_2 \), where \( M_1 \) and \( M_2 \) are shown below. The initial state of \( M_1 \) is \( s_0 \), the initial state of \( M_2 \) is \( t_0 \):

**Kripke structure \( M_1 \):**

- \( s_0: \{b\} \)
- \( s_1: \{b\} \)
- \( s_2: \{c\} \)
- \( s_3: \{a\} \)
- \( s_4: \{a\} \)

**Kripke structure \( M_2 \):**

- \( t_0: \{b\} \)
- \( t_1: \{b\} \)
- \( t_2: \{b\} \)
- \( t_3: \{c\} \)
- \( t_4: \{c\} \)
- \( t_5: \{a\} \)

(4 points)
(b) Consider the following Kripke structure $M$:

```
M,

s0: {a}  s1: {a, c}

s2: {a}  s3: {a, b, c}  s4: {a, b, c}
```

For each of the following formulae $\varphi$,

i. check the respective box if the formula is in CTL, LTL, and/or CTL*, and

ii. list the states $s_i$ on which the formula $\varphi$ holds; i.e. for which states $s_i$ do we have $M, s_i \models \varphi$?

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>CTL</th>
<th>LTL</th>
<th>CTL*</th>
<th>States $s_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F(c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AX(c)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E[(a) U (b)]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G(a \land b)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$EG(a)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(5 points)
(c) LTL tautologies

Prove that the following formulas are tautologies, i.e., they hold for every Kripke structure $M$ and every path $\pi$ in $M$, or find a Kripke structure $M$ and path $\pi$ in $M$, for which the formula does not hold and justify your answer.

i. $((\neg a \mathcal{U} b) \mathcal{U} \neg c) \leftrightarrow (\neg a \mathcal{U} (b \mathcal{U} \neg c))$

ii. $(\mathsf{FG} a) \Rightarrow \mathsf{G}(a \mathsf{V} \mathsf{XF} a)$

(6 points)