1. (50) (Fixed Point Semantics) The function $F$ is defined as follows:

\[ F = \lambda g.\lambda n.(\text{if } (n = 1) \text{ then } 2 \text{ else if } \text{even}(n) \text{ then } n + g(n/2) \text{ else } g(2n - 3)) \]

(a) Please list the functions $F \perp$, $F^2 \perp$, $F^3 \perp$, in their simplest form.

**Answer:**
- $F \perp n = \text{if } n = 1 \text{ then } 2 \text{ else } \perp$;
- $F^2 \perp n = \text{if } n = 1 \lor n = 2 \text{ then } 2n \text{ else } \perp$;
- $F^3 \perp n = \text{if } n = 1 \lor n = 2 \lor n = 4 \text{ then } 2n \text{ else } \perp$.

(b) Please provide the function $F^i \perp$ and prove by induction on $i$ that your $F^i \perp$ is correct.

**Answer:**

$F^i \perp n = \text{if } n = 2^j (0 \leq j < i) \text{ then } 2n \text{ else } \perp$.

$F^0 \perp = \perp$ is the base case when $n = 0$.

For the inductive case,

\[ F^{i+1} \perp n = \text{if } (n = 1) \text{ then } 2 \text{ else if } \text{even}(n) \text{ then } n + F^i \perp (n/2) \text{ else } F^i \perp (2n - 3) \]

\[ = \text{if } n = 2^j (0 \leq j \leq i) \text{ then } 2n \text{ else } \perp . \]

(c) Please find the least fixed point $f_1$ of $F$.

**Answer:** $f_1 n = \text{if } n = 2^j (0 \leq j) \text{ then } 2n \text{ else } \perp$.

(d) Please find two other different fixed points $f_2$ and $f_3$ of $F$ such that $f_1 \subseteq f_2 \subseteq f_3$, where $\subseteq$ is the partial order on functions as given in the textbook.

**Answer:**

- $f_2 n = \text{if } n = 0 \lor n = 2^j (0 \leq j) \text{ then } 2n \text{ else } \perp$.
- $f_3 n = \text{if } n = 0 \lor n = 2^j \lor n = 3 \times 2^j (0 \leq j) \text{ then } 2n \text{ else } \perp$.

We can verify that $F f_i = f_i$ for $i = 1, 2, 3$ and $f_1 \subseteq f_2 \subseteq f_3$. 
2. (50) **(Denotational Semantics)** We like to add into Wren two features of C language.

(a) One is the C-style conditional expression:

\[
\text{Expression ::= ... | Expression? Expression : Expression}
\]

Please provide denotational semantics for the conditional expression in Wren and use your definition to prove the semantic equivalence of the following two commands:

\[
m := e_1 ? e_2 : e_3;
\]

and

\[
\text{if } e_1 \text{ then } m := e_2 \text{ else } m := e_3;
\]

**Answer:** Please see the answer to Problem 9.3.1 b).

(b) The other is the operators `++` and `--`: `++ i` in an expression will return one plus the value of `i`; `i++` will return the value of `i`. In both cases, the value of `i` will be increased by one. The meaning for `--` is similar.

Please modify Wren’s denotational semantics to handle `++` and `--` (both before and after an integer identifier).

**Answer:** The store needs to be updated and the updated store should be returned by `evaluate`. Please see the answer to Problem 9.3.1 c).