Assume that DrRacket has been started up in the Intermediate Student with lambda language, and that the following commands have been run. The command \((\text{cons? } L)\) returns \text{true} if \(L\) is a list, \text{false} otherwise. The contract \(; \text{np : list/number -> number}\) says that the function \(\text{np}\) consumes a number or a list, and produces a number.

\[
; \text{np : list/number -> number}
\]

\[
\text{(define (np L)}
\]

\[
; \text{(cond}
\]

\[
[(\text{cons? } L) (\text{apply } \ast (\text{map np } L))]
\]

\[
[\text{else } L)])
\]

\[
\text{(define L1 3)}
\]

\[
\text{(define L2 (list 1 2 3))}
\]

\[
\text{(define L3 (list (list 1 2) 3 (list 4 5)))}
\]

For full marks, describe or draw what is produced by the following commands.

\[
(\text{np L1}) ; \text{in the definition of np, replace L by the value of L1, that is 3}
\]

\[
(\text{np L2}) ; \text{in the definition of np, replace L by the value of L2, that is (list 1 2 3)}
\]

\[
\text{; use what you know about (np L1)}
\]

\[
(\text{np L3}) ; \text{in the definition of np, replace L by the value of L3,}
\]

\[
\text{; that is (list (list 1 2) 3 (list 4 5))}
\]

\[
\text{; use what you know about (np L1) and (np L2)}
\]