(a) Given list \( P \), find list \( L \) such that for every index \( n \) of list \( P \), \( L_n \) is the length of the longest list that is both a proper prefix and a proper suffix of \( P[0..n+1] \). Here is a program to find \( L \).

\[
\begin{align*}
A & \equiv i := 0. \quad L := \lceil #P \rceil \cdot j := 1. \quad B \\
B & \equiv \text{if } j \geq #P \text{ then } \text{ok} \ \text{else} \ C. \quad L := j \rightarrow i \mid L. \quad j := j + 1. \quad B \ \text{fi} \\
C & \equiv \text{if } P_i = P_j \text{ then } i := i + 1 \\
& \quad \text{else if } i = 0 \text{ then } \text{ok} \\
& \quad \quad \text{else } i := L(i - 1). \quad C \ \text{fi} \\
\end{align*}
\]

Find specifications \( A \), \( B \), and \( C \) so that \( A \) is the problem and the three refinements are theorems.

(b) Given list \( S \) (subject), list \( P \) (pattern), and list \( L \) (as in part (a)), determine if \( P \) is a segment of \( S \), and if so, where it occurs. Here is a program.

\[
\begin{align*}
D & \equiv m := 0. \quad n := 0. \quad E \\
E & \equiv \text{if } m = #P \text{ then } h := n - #P \ \text{else} \ F \ \text{fi} \\
F & \equiv \text{if } n = #S \text{ then } h := \infty \\
& \quad \text{else if } P_m = S_n \text{ then } m := m + 1. \quad n := n + 1. \quad E \\
& \quad \quad \text{else } G \ \text{fi} \\
G & \equiv \text{if } m = 0 \text{ then } n := n + 1. \quad F \ \text{else} \ m := L(m - 1). \quad G \ \text{fi} \\
\end{align*}
\]

Find specifications \( D \), \( E \), \( F \), and \( G \) so that \( D \) is the problem and the four refinements are theorems.