1 Question 1

Write a function \( \text{(list-pos \los \text{item})} \) that consumes a \((\text{listof Str})\) and a \(\text{Str}\) and returns the position of the first occurrence of \text{item} in \los. \los is non-empty and \text{item} is guaranteed to be in it. The first item is in position 0. For example:

\[
(\text{list-pos (list "a" "b" "c") "b")} \Rightarrow 1
\]

Use recursion! Do not use \text{map}, \text{foldr}, or \text{filter}.

2 Question 2

Write a function \( \text{(multiple \text{s} \text{n})} \) that returns the \text{Str} created by appending \text{n} copies of \text{s}. For example:

\[
(\text{multiple "abc" 3}) \Rightarrow "abcabcabc"
\]

Use recursion! Do not use \text{map}, \text{foldr}, or \text{filter}.

3 Question 3

Create a function \( \text{(next-list \text{L} \text{target})} \) that consumes a \((\text{listof Any})\) and an \(\text{Any}\), and returns the item in the list that appears after \text{target}. \text{next-list} requires that \text{target} is present in \text{L}, and that it is not the last item in the list. For example:

\[
(\text{next-list (list 1 2 3) 2}) \Rightarrow 3
\]

If \text{target} appears more than once in \text{L}, consider the first occurrence.

\[
(\text{next-list (list 2 4 5 7 8 7 3 6 7 9) 7}) \Rightarrow 8
\]

Use recursion! Do not use \text{map}, \text{foldr}, or \text{filter}.
Write a function (difference lon) that consumes a (listof Num) and returns a list of differences between adjacent pairs in lon.
You may assume that lon is not empty.
For example:
(differences (list 42)) => ()
(differences (list 25 16 9 1 4)) => (list 9 7 8 -3)

Use recursion! Do not use map, foldr, or filter.