CS115 – Lab 7: Recursion

Spring 2019

Question 1: Locating a Value

Exercise
Write a function \((\text{list-pos } \text{L item})\) that consumes a \((\text{listof Str})\) and a \(\text{Str}\) and returns the position of the first occurrence of \(\text{item}\) in \(\text{L}\).

\(\text{L}\) is non-empty and \(\text{item}\) is guaranteed to be in it. The first item is in position 0.

For example:
\[(\text{list-pos (list "Odlaw" "Wenda" "Waldo" "Woof") "Waldo") } \Rightarrow 2\]

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

Question 2: Copying Strings

Exercise
Write a function \((\text{copy } s n)\) that returns the \(\text{Str}\) created by appending \(n\) copies of \(s\).

For example:
\[(\text{copy "that" 3}) \Rightarrow "thatthatthat"\]

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

Question 3: Building Stairs

Exercise
Write a function \((\text{stair } n)\) that returns a \((\text{listof Str})\) of length \(n\) where the first item is "X", and every subsequent item contains one more "X".

For example,
\[(\text{check-expect (stair 4)}\]
\[(\text{list "X" "XX" "XXX" "XXXX")})\]

Hint
You may use \text{copy} as a helper function.

Use recursion! Do not use \text{map}, \text{foldr}, \text{filter}, or \text{range}.
Question 4: Hide & Seek

Create a function (next-list L target) that consumes a (listof Any) and an Any, and returns the item in the list that appears after target.

If target is not present in L or is the last item in L, the function returns #false.

For example:

(next-list (list 1 2 3) 2) => 3
(next-list (list 1 2 3) 7) => #false
(next-list (list 1 2 3) 3) => #false

If target appears more than once in L, consider the first occurrence.

(next-list (list 2 45 7 8 7 3 6 7 9) 7) => 8

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