

<

Previous















Next

>

Question 2

 [Bookmark this page](#)

You are asked to design a function that numbers a list of strings by inserting a number and a colon before each element of the list based on its position. So for example:

```
(number-list (list "first" "second" "third")) would produce (list "1: first" "2: second" "3: third")
```

Here is a partially completed design of this function

```
;; (listof String) -> (listof String)
;; append each string's position in the list to the front of the string to number the list
(check-expect (number-list empty) empty)
(check-expect (number-list (list "first" "second" "third"))
              (list "1: first" "2: second" "3: third"))
(define (number-list los) los) ;stub
```

Question 2

1/1 point (graded)

Does `number-list` require a context-preserving accumulator?

☐ no, we can complete the design of `number-list` using a standard list template

☐ yes, we need to keep track of the previous element of the list

☒ yes, we need to keep track of the current element's position in the list

☐ yes, we need to keep track of the colon and space



Explanation

In order to add the number in front of each element, we must know its position in the list.

Submit

 Show Answer

 Answers are displayed within the problem

<

Previous

Next

>



edX

About

[edX for Business](#)

Legal

[Terms of Service & Honor Code](#)

[Privacy Policy](#)

[Accessibility Policy](#)

Connect

[Blog](#)

[Contact Us](#)

[Help Center](#)



© 2020 edX Inc. All rights reserved.
| 深圳市恒宇博科技有限公司 粤ICP备17044299号-2