

< Previous



Next >

Questions 1-4

[Bookmark this page](#)

Questions 1-2

2/2 points (graded)

Let's complete the design of `insert-string` that we saw in the previous video.

```
;; String ListOfString -> ListOfString
;; insert string in correct place in los (in ascending order)
;; ASSUME: los is already sorted

(define (insert-string str los) los) ;stub
```

What is the result of the following check-expects?

```
(check-expect (insert-string "Apple" empty) ____)
```

☐ empty

☐ "Apple"

☒ (cons "Apple" empty)



Explanation

Inserting "Apple" into an empty list should give us the one element list `(cons "Apple" empty)`

```
(check-expect (insert-string "Sally" (cons "Apple" (cons "Systematic" empty))) ____)
```

☐ (cons "Sally" (cons "Apple" (cons "Systematic" empty)))

☒ (cons "Apple" (cons "Sally" (cons "Systematic" empty)))

☐ (cons "Systematic" (cons "Sally" (cons "Apple" empty)))



Explanation

"Sally" should go after "Apple" and before "Systematic"

Submit

Show Answer

Answers are displayed within the problem

Questions 3-4

2/2 points (graded)

The function body is similar to `insert` that you saw in this video. You may want to look at `string>=?` to make sure you understand what it does?

```
(define (insert-string str los)
  (cond [(empty? los) (cons str empty)]
        [else
         (if (string>=? str (first los))
             A1
             A2)]))
```

What should go in place of A1?

☐ (cons str los)

☒ (cons (first los) (insert-string str (rest los)))

☐ (insert-string str (rest los))

☐ (insert-string str (rest los))



Explanation

If `str` comes alphabetically after the first item of the list, we should insert `str` somewhere in the rest of the list and keep the first item in front of that.

What should go in place of A2?

☒ (cons str los)

☐ (cons (first los) (insert-string str (rest los)))

☐ true

☐ false



Explanation

If `str` comes alphabetically before the first item in the list, we should insert `str` at the beginning at the list.

Submit

Show
Answer

Answers are displayed within the problem

[< Previous](#) [Next >](#)

© 2020 edX Inc. All rights reserved.



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