# Questions 1-3

Bookmark this page

Consider the following Data Definition for ListOfNumber:

```
;; ListOfNumber is one of:
;; - empty
;; (cons Number ListOfNumber
;; interp. a list of numbers
(define LON1 empty)
(define LON2 (cons 1 empty))
(define LON3 (cons 2 (cons 1 empty)))
#;
(define (fn-for-lon lon)
  (cond [(empty? lon)(...)]
        [else
          (... (first lon)
               (fn-for-lon (rest lon)))]))

;; Template Rules Used:
;; - one of: 2 cases
;; - atomic distinct: empty
;; - compound: (cons Number ListOfNumber)
;; - [coming soon]
```

We would like to design a function that consumes a list of numbers and produces true if that list contains a negative number.

## Questions 1

1/1 point (graded)

Here is the signature and purpose for the function that produces true if the list contains a negative number.

```
;; ListOfNumber -> Boolean
;; produce true if a lon contains a negative number
```

What is the result of the following tests?

```
(check-expect (contains-negative? (cons 1 empty)) _____)
```

false ✔ **Answer:** false

**Explanation**
The list has only 1 element and it is positive.

Submit Show Answer

Answers are displayed within the problem

## Question 2

1/1 point (graded)

```
(check-expect (contains-negative? (cons 1 (cons -1.5 empty))) _____)
```

true ✔ **Answer:** true

**Explanation**
The second element of the list is negative.

Submit Show Answer

Answers are displayed within the problem

## Question 3

1/1 point (graded)

```
(check-expect (contains-negative? empty) _____)
```

false

✔ **Answer:** false

**Explanation**
The list is empty, so it contains no negative numbers.

Submit

Show Answer

Answers are displayed within the problem

Previous Next

Some Rights Reserved



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