Question 1

Question 1

1/1 point (graded)

Which of the following functions is a correct implementation of the search list case of a backtracking search?

```
(define (find-entry--loe e loe)
  (cond [(empty? e) false]
        [else
         (if (not (false? (find-entry--entry e (first loe))))
             false
             (find-entry--loe e (rest entry)))]))
```

```
(define (find-entry--loe e loe)
  (cond [(empty? e) 0]
        [else
         (+ (find-entry--entry e (first loe))
            (find-entry--loe e (rest entry)))]))
```

```
(define (find-entry--loe e loe)
  (cond [(empty? e) false]
        [else
         (if (not (false? (find-entry--entry e (first loe))))
             (find-entry--entry e (first loe))
             (find-entry--loe e (rest entry)))]))
```

```
(define (find-entry--loe e loe)
  (cond [(empty? e) false]
        [else
         (cons (find-entry--entry e (first loe))
               (find-entry--loe e (rest entry)))]))
```

✔

Explanation

This is backtracking search, because if looking in the first branch produces `false` then the search fails and backtracks to look into the rest of the elements in the branch. Otherwise the function produces the result of searching the first branch.

Submit

Answers are displayed within the problem