Course › 7b: Local › Avoiding Recomputation › Questions 1-2

Questions 1-2

Question 1

1/1 point (graded)

In the practice problem find-person-starter.rkt we asked you to use local to improve your functions performance as the tree gets larger. Would the following be a correct use of local for that purpose?

```
(define (find--lop n lop)
  (local [(define try (find--person n (first lop)))]
    (cond [(empty? lop) false]
          [else
           (if (not (false? (find--person n (first lop))))
               (find--person n (first lop))
               (find--lop n (rest lop)))])))
```

no ✓ **Answer:** no

Explanation

In this case the local has not been moved to surround the nearest expression that encloses all occurrences of the repeated computation. That would be around the if. Instead it moved outside of the cond answer the if appeared in. Try running it with this definition and see what happens. Step it carefully to be sure you understand why.

Submit

Answers are displayed within the problem

Question 2

1/1 point (graded)

Would this version that doesn't use local work?

```
(define try (find--person n (first lop)))

(define (find--lop n lop)
  (cond [(empty? lop) false]
        [else
         (if (not (false? try))
             try
             (find--lop n (rest lop)))]))
```

no ✓ **Answer:** no

Explanation

The paramaters n and `lop` are not defined outside of `find--lop`, so this will not work either.

Submit

Answers are displayed within the problem