ComS 342
Recitation 2, 10:00 Tuesday
Homework 5

Sean Gordon

October 18, 2019

1a.

```
(define helper (
    lambda (chr lst)

        (//Look through this string for the char
        if (null? lst)
            (list)
            (
            if (= (car lst) chr) //If this is the char we're looking for...
                (list chr)              //Return some non-null list
                (helper chr (cdr lst))  //Otherwise keep looking
))))


(define Find (
    lambda(chr lst)
        (
        if (null? lst)
            (list)
            ( // If this string is not the one...
            if (null? (helper chr (car lst)))
                (Find chr (cdr lst))  // Return the same thing but with the
                                      // next string
                (car lst)             // Otherwise return this string
))))
```

```
(define helper (lambda (chr lst)(if (null? lst)(list)
   (if (= (car lst) chr)(list chr)(helper chr (cdr lst))))))
(define Find (lambda(chr lst)(if (null? lst)(list)
   (if (null? (helper chr (car lst)))(Find chr (cdr lst))(car lst)))))
```

1b.

```
(define helper1 (
    lambda (curr rest)
        (
            if (null? curr)
            (Concatenate rest)
            (cons (car curr)(helper1 (cdr curr) rest))
        )
))

(define Concatenate (
    lambda(lst)
        (
        if (null? lst)
            (list)
            (
            if (null? (car lst))
                (Concatenate (cdr lst))
                (helper1 (car lst) (cdr lst))
            )
        )
))
```

(define helper1 (lambda (curr rest)(if (null? curr)(Concatenate rest)(cons (car curr) (helper1 (cdr curr) rest)))))
(define Concatenate (lambda(lst)(if (null? lst)(list)(if (null? (car lst)) (Concatenate (cdr lst))(helper1 (car lst) (cdr lst))))))

2.

```
(define rest (
    lambda (x l)
        (
        if (null? l)
            (list)
            (
            if (= (car l) x)
                (cdr l)
                (cons (car l) (rest x (cdr l)))
            )
        )
))

(define helper(
    lambda (x lst)
        (
        cons x (Shuffle (rest x lst))
        )
))

(define Shuffle(
    lambda (lst)
        (
        if (null? lst)
            (list)

            (
            helper(Random lst) lst
            )
        )
))
```

```
(define rest (lambda (x l)(if (null? l) (list)(if (= (car l) x)(cdr l)
  (cons (car l) (rest x (cdr l)))))))
(define Shuffle(lambda (lst)(if (null? lst)(list)(helper (Random lst) lst))))
(define helper(lambda (x lst)(cons x (Shuffle (rest x lst)))))
```