

CSSE 304 Day 10

0. Curried versions of `member?` and `map`:

```
(define member? ;curried version
  (lambda (item ls)
    ((member?-c item) ls)))
```

```
(define map ;curried version
  (lambda (proc ls)
    ((apply-to-all proc) ls)))
```

Write the following four list procedures in the style, so we can notice and exploit the similarities and differences

1. (define member?-c

2. (define apply-to-all

3. (define list-sum

4. (define list-product

5. Write a procedure that abstracts the list processing in the above procedures:

```
(define list-recur
```

6. Use `list-recur` to define the procedures from 1-4:

```
(define member?-c
```

```
(define apply-to-all
```

```
(define list-sum
```

```
(define list-prod
```

7. Use `list-recur` to define `length`:

```
(define length
```

8. Before the next class meeting:

- a. Think of other procedures that can be written using `list-recur`. Write and test them.
- b. Of the five procedures that we wrote today using `list-recur`, one is a bad idea. Which one, and why?