

CS115 Abstract List Functions Exercise:

What do the following functions produce? (try to do this without using Dr. Racket)

Question 1:

```
(map sqr (list 1 2 3 4))
```

Question 2:

```
:: Helper Function:  
;; remainder-of-seven: Nat -> Nat  
(define (remainder-of-seven n)  
  (remainder n 7))
```

```
(map remainder-of-seven (list 0 6 7 14 23))
```

Question 3:

```
:: Helper Function:  
;; inbetween?: Num Num  
(define (inbetween? n)  
  (or (and (< n 5) (>= n 3)) (and (>= n 10) (< n 20))))
```

```
(filter inbetween? (list 5 4 11 9 3))
```

Question 4:

```
:: Defined Constant:  
(define foodlist (list "apples" "oranges" "avocados" "bananas" "aoli"))
```

```
:: Helper Function  
;; is-first-a?: Str -> Bool  
(define (is-first-a? s)  
  (equal? "a" (substring s 0 1)))
```

```
(filter is-first-a? foodlist)
```

Question 5:

```
;; Structure Definition:
(define-struct pasta-sauce (brand type quantity))
;; is a (make-pasta-sauce Str Sym Num) where
;; brand is the brand of the pasta sauce,
;; type is the kind of the pasta sauce, and
;; quantity is the quantity of the pasta sauce in ounces.

(define sauce1 (make-pasta-sauce "newman" 'tomato 24))
(define sauce2 (make-pasta-sauce "ragu" 'tomato 66))
(define sauce3 (make-pasta-sauce "classico" 'alfredo 14))

(foldr + 0 (map pasta-sauce-quantity (list sauce1 sauce2 sauce3)))
```

Question 6:

```
;; Helper Function
;; divisors?: Nat Nat Nat -> Bool
(define (divisors? n a)
  (cond
    [(= n a) false]
    [(zero? (remainder n a)) true]
    [else (divisors? n (add1 a))]))

;; Helper Function
;; is-prime?: Nat -> Bool
(define (is-prime? n)
  (cond
    [(<= n 1) false]
    [(= n 2) true]
    [else (not (divisors? n 2))]))

(foldr * 1 (filter is-prime? (list 1 2 3 4 5 6 7 8 9 10)))
```