Question 1. Assume the following two definitions have been entered/run ...

```scheme
(define (add-five n) (+ 5 n))
(define (cherry pie) "cherry")
```

... and show the following steps (when an error occurs you can write it down as just "error") ...

```scheme
(step (map add-five (list 12 43 65)))
```

```scheme
(step (cherry "apple"))
```

```scheme
(step (cherry "cherry"))
```

```scheme
(step (cherry (list 123 (circle 10))))
```

```scheme
(step (cherry (- "apple")))
```

```scheme
(step (map cherry (list "one" 2 "three")))
```

```scheme
(step (combine text-join (map cherry (range 3))))
```
Question 2. Assume the following three definitions have been entered/run ...

```lisp
(define (pie? a-food) (same? (first a-food) "pie"))
(define (two? n) (same? n 2))
(define (full-name a-food) (text-join (second a-food) " " (first a-food)))
```

... and show the following steps, but you may omit the underlining ...

```lisp
(step (map first (list (list "roll" "unagi")))
      (list "pie" "apple")
      (list "roll" "tuna")))
```

```lisp
(step (map pie? (list (list "roll" "unagi"))
               (list "pie" "apple")
               (list "roll" "tuna")))
```

```lisp
(step (map second (reverse (list (list "roll" "unagi")
                                (list "pie" "apple")
                                (list "roll" "tuna")))))
```

```lisp
(step (second (map reverse (list (list "roll" "unagi")
                                (list "pie" "apple")
                                (list "roll" "tuna")))))
```
(step (map two? (map length (list (list "roll" "unagi")
                                (list "pie" "apple")
                                (list "roll" "tuna")))))

(step (map full-name (list (list "roll" "unagi")
                            (list "pie" "apple")
                            (list "roll" "tuna")))))

; Question 3. Assume the following two definitions have been entered/run ...
(define (add-to-all constant numbers)
  (map ((fun n) (+ constant n))
       numbers))
(define (n-of n thing)
  (map ((fun _ ) thing)
       (range n)))

; ... and for each of the following show ONLY the first step, which substitutes the arguments into the body ...
(step (add-to-all  5 (list 12 43 65)))

(step (add-to-all 10 (list 12 43 65)))

(step (add-to-all 78 (list 12 43 65)))

(step (n-of 10 0))

(step (n-of 20 "strawberry"))