

### Exercise 3.13.

Consider the following `make-cycle` procedure, which uses the `last-pair` procedure defined in exercise 3.12:

```
(define (make-cycle x)
  (set-cdr! (last-pair x) x)
  x)
```

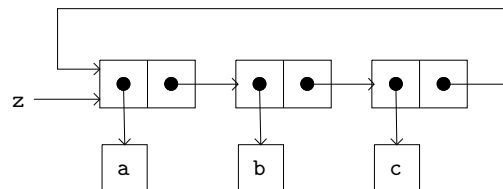
Draw a box-and-pointer diagram that shows the structure `z` created by

```
(define z (make-cycle (list 'a 'b 'c)))
```

What happens if we try to compute `(last-pair z)`?

**Answer.**

Figure 1 illustrates the structure `z` created by evaluating `(define z (make-cycle (list 'a 'b`




**Figure 1.** The list formed by `(define z (make-cycle (list 'a 'b 'c)))`

`'c)))`.

Trying to evaluate `(last-pair z)` will draw the interpreter busy in looping the cycle we've just created without termination.

---

\*. Creative Commons  2013, Lawrence X. Amlord (颜世敏, aka 颜序).