* 연습 문제 *

다음을 LISP 에서 수행해 보고 각각의 primitive의 기능을 실습.

1. car 또는 first

```
> (car '(one two three))
> (car '((1 2) 3 4 (5 6)))

> (first '(fast computers are nice))
> (first '(a b c))
```

2. cdr 또는 rest

```
> (cdr '(one two three))
> (cdr '((1 2) 3 4 (5 6)))
> (cdr '(a))

> (rest '(a (b c) d))
> (rest '(a b c))
> (rest '((a b) (c d)))
> (rest '(a))
> (rest ())
```

3. cxxxxr

```
> (setq layered-list '((second layer) first (second (third)) first))
> (car layered-list)
> (cdr layered-list)
> (car (cdr layered-list))
> (cadr layered-list)
> (cdr (car (cdr (cdr layered-list))))
> (cdaddr layered-list)
```

4. setf

```
> (setf ab-list '(a b))
> ab-list

> (setf ab-list '(a b) xy-list '(x y))
> ab-list
> xy-list
```

5. cons

```
> (cons 'one '(two three))
> (cons '(1 2) '(3 4 (5 6)))
> (setq some-num (* 4 3))
> some-num
> (setq my-list (cons 'a '(b c)))
> my-list
> (cdr my-list)
> (cons some-num my-list)

> (cons 'a 'b)
```

6. append

```
> (append '(a b c) '(d e f))
> (append '(a b c) '(d) '(e))
> (append '(a) 'b)
> (append '(a) '((b) (c)))
> (append '(a b c) '(d e f) 'g)
> (append '(one (two)) '(((three))))
```

7. list

```
> (list '(a b c) '(d e f) 'g)
> (list '(one (two)) '(((three))))
> (list 'a 'b)
> (list '(a b c) '(d e f))
> (list 'a 4 (+ 2 3))

> (list* 'a 'b 'c)
> (list* 'a '(b c))
> (list* '(a b))
```

8. nthcdr

```
> (nthcdr 3 '(0 1 2 3 4 5))
> (nthcdr 0 '(a b c))
> (nthcdr 2 '(a (b c) (d e)))
> (nthcdr 1 '(a b c))
> (nthcdr 3 '(a b c))
```

9. butlast

```
> (butlast '(a b c))
> (butlast '(a))
> (butlast '())
> (butlast '((a b) (c d) (e f)) 2)
```

10. last

```
> (last '(1 2 3))
> (last '(1 (2) (3 4)))
> (last '(a b c))
> (last '(a b c.d))
> (last ())
> (last '((a b) (c d)))
```

11. nth

```
> (nth 4 '(a b c d e))
> (nth 2 '(a (b c) (d e)))
> (nth 0 '(a b c))
> (nth 3 '(a b (c d) (e f)))
> (nth 2 '(a b))
```

12. first, second, ..., tenth

```
> (first '((a b) (c d)))
> (second '((a b) (c d)))
> (third '(a b c))
```

13. length

```
> (length '(a b))
> (length '((a b) (c d)))
```

14. list-length

```
(list-length '(a b c))
(list-length '((a b) (c d)))
(list-length ())
```

15. reverse

```
(reverse '(a b c))
(reverse '((a b) (c d)))
(reverse "abc")
```

16. revappend

```
(revappend '(a b c) '(1 2 3))
(revappend '(a (b c) (d e)))
(revappend '(a (b c) (d e)) '(1 2 3))
```

17. make-list

```
(make-list 4)
(make-list 2 :initial-element 'ho)
```

18. endp

```
(endp ())
(endp '(a b c))
(endp (cdr '(a)))
```

19. listp

```
(listp '(a b c))
(listp 'a)
(listp nil)
```

* 문제 1 *

아래 실행 결과들을 output file로 출력하려면...

```
> (first (rest (first (rest '((a b) (c d) (e f))))))
> (first (first '(rest (rest ((a b) (c d) (e f))))))
> (first '(first (rest (rest ((a b) (c d) (e f))))))
> (append '(a b c) '())
> (append '((car chevrolet) (drink coke)) (reverse '((car chevrolet) (drink coke))))
> (list '(a b c) '())
> (cons '(a b c) '())
> (cons (first nil) (rest nil))
```

* 문제 2 *

아래와 같은 형태의 list에서 Harry를 얻기 위한 방법? (car, cdr 등의 함수를 이용함)

1) (apples in (Harry has a backyard))
2) (apples and Harry)
3) (((apples) and ((Harry))) in his backyard)