Allatoms: version 1

```
(defun a1 (list)
   (cond ( (null list) nil )
         ( (eq (length list) 1) list )
         ( t  (cons (a1 (car list))
                    (a1 (cdr list))) )
   )
)
```

```
USER(17): (a1 '((b c)) )
((B C))
```

```
USER(18): (a1 '(a (b c) d) )
Error: attempt to take the length of a
non-sequence: A
```


Allatoms: version 2

```
(defun a2 (list)
   (cond ( (null list) nil )
         ( (eq (length list) 1) list )
         ( t  (append
                 (cond ( (atom (car list))
                             (list (car list)) )
                       ( t (a2 (car list)) )
                 )
                 (a2 (cdr list))) )
   )
)
```

```
USER(25): (a2 '(a (b c) d) )
(A B C D)

USER(27): (a2 '((a b c) ((d)) (e (f (g)) h)))
(A B C (D) (E (F (G)) H))

USER(30): (a2 '((b c)) )
((B C))
```

Allatoms: version 3

```
(defun a3 (list)
   (cond ( (null list) nil )
         ( t  (append
                 (cond ( (atom (car list))
                             (list (car list)) )
                       ( t (a3 (car list)) )
                 )
                 (a3 (cdr list))) )
   )
)
```

```
USER(33): (a3 '((b c)) )
(B C)

USER(34): (a3 '(a (b c) d) )
(A B C D)

USER(35): (a3 '((a b c) ((d)) (e (f (g)) h)))
(A B C D E F G H)
```


Countatoms

```
(defun countatoms (list)
   (cond ( (null list) 0 )
         ( t (+
                (cond ( (atom (car list)) 1 )
                      ( t (countatoms (car list)) )
                )
                (countatoms (cdr list))) )
   )
)
```