

```
nick@locutus] more listatoms
(defun listatoms (x)
  (prog (res)
    loop (cond ((null x) (return res))
            ((atom (car x)) (setq res (cons (car x) res)))
            (t (setq res (append (listatoms (car x)) res))))
    (setq x (cdr x))
    (go loop)))
```

```
nick@locutus] gcl
GCL (GNU Common Lisp) Version(2.4.0) Tue Apr 30 13:15:51 ADT 2002
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
```

```
>(load "listatoms")
```

```
Loading listatoms
Finished loading listatoms
T
```

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

```
(D C B A)
```

```
>(listatoms '(a ((b) c ((d))) e))
```

```
(E D C B A)
```

```
>(bye)
```

```
nick@locutus] more flatten
(defun flatten (l)
  (cond ((null l) nil)
        ((atom l) (list l))
        (t (append (flatten (car l))
                    (flatten (cdr l))))))
```

```
(defun flat (x)
  (cond ((atom x) (list x))
        (t (mapcan 'flat x))))
```

```
nick@locutus] gcl
GCL (GNU Common Lisp) Version(2.4.0) Tue Apr 30 13:15:51 ADT 2002
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
```

```
>(load "flatten")
```

```
Loading flatten
Finished loading flatten
T
```

```
>(flatten '(a ((b))) (c (d (e (((f)) g) h) i) j)))
```

```
(A B C D E F G H I J)
```

```
>(flat '(a ((b))) (c (d (e (((f)) g) h) i) j)))
```

```
(A B C D E F G H I J)
```

```
>(bye)
```