The following example returns the argument if it is an atom, NIL if it is an empty list and the first element if the argument is a list.

```
(DEFUN FIRST (X)
        (COND
                ((ATOM X) X)
                ((NULL X) NIL); useless
                (T (CAR X))
        )
)
```

The following example returns the maximum values of the two arguments.

```
(DEATH MAX (X Y)
        (COND
                ((> X Y) X)
                (T Y)
        )
)
```

The following example returns the last item in a list, superficially.

```
(DEFUN LAST (X)
        (COND
                ((ATOM X) X)
                ((NULL (CDR X)) (CAR X))
                (T (LAST (CDR X)))
        )
)
```

The following example rewrites CAR to return NIL if the argument is atom and does not produce an error message.

```
(DEFUN XCAR (X)
        (COND
                ((ATOM X) NIL)
                (T (CAR X))
        )
)
```

The following example shows a possible definition for the APPEND function.

```
(DEFUN APPEND (L1 L2)
        (COND
                ((NULL L1) L2)
                (T (CONS (CAR L1)) (APPEND (CDR L1) L2))); copy the L1 list
        )
)
```

The following example shows a possible definition for the MEMBER function.

```
(DEFUN MEMBER (ELEM LIST)
        (COND
                ((ATOM LIST) NIL)
                ((EQUAL ELEM (CAR LIST)) LIST)
                (T (MEMBER ELEM (CDR LIST)))
        )
)
```

A possible implementation of the REMOVE function is:

```
(DEFUN REMOVE (e l)
        (COND
                ((ATOM l) l)
                ((EQL (CAR l) e) (REMOVE e (CDR l)))
                (T (CONS (CAR l) (REMOVE e (CDR l))))
        )
)
```

A possible implementation of the SUBST function is:

```
(DEFUN SUBST (e1 e2 e3)
        (COND
                ((EQUAL e2 e3) e1)
                ((ATOM e3) e3)
                (T (CONS (SUBST e1 e2 (CAR e3)) (SUBST e1 e2 (CDR e3))))
        )
)
```