Lisp Homework #2: LOOKUP

You are to write a LISP function, LOOKUP which works as follows –

```
(lookup ‘a ‘((a b) (c d)) => b

(lookup ‘e ‘((a b) (c d)) => nil

(lookup ‘a ‘((b a) (c d)) => nil

(lookup ‘e  ‘((a b) (c d) (e f)) => f

(lookup ‘c ‘((a b) (c (d e f)) => (d e f)
```

The function will be passed an atom and a list of pairs. If the atom matches the first element of the pair, then it should return the second element. If the atom is not found, then it should return nil. Note that the pairs may contain elements which are not atoms.

Turn this in via EASEL.

There is a built in function which does this operation. You cannot use it. Build it out of basic CAR, CDR, CONS, LIST etc. functions.

This function will be tested on the Taz2 CLISP implementation.