```
Last login: Wed Aug  1 08:11:43 on ttys000
Josephs-Air-2:~ JosephHayes$ cd desktop/61a/projects/scheme
Josephs-Air-2:scheme JosephHayes$ ls
__pycache__       proj04.ok        scheme_tokens.py
buffer.py         questions.scm        tests
images            scheme.py        tests.scm
mytests.rst       scheme_builtins.py   ucb.py
ok          scheme_reader.py
Josephs-Air-2:scheme JosephHayes$ python3 ok -q 07 -u
=====================================================================
Assignment: Project 4: Scheme Interpreter
OK, version v1.13.11
=====================================================================


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Unlocking tests

At each "? ", type what you would expect the output to be.
Type exit() to quit


---------------------------------------------------------------------
Problem 7 > Suite 1 > Case 1
(cases remaining: 6)

Q: What is the structure of the expressions argument to do_quote_form?
Choose the number of the correct choice:
0) Pair(A, nil), where:
        A is the quoted expression
1) [A], where:
        A is the quoted expression
2) Pair('quote', Pair(A, nil)), where:
        A is the quoted expression
3) A, where:
        A is the quoted expression
? 2
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair(A, nil), where:
        A is the quoted expression
1) [A], where:
        A is the quoted expression
2) Pair('quote', Pair(A, nil)), where:
        A is the quoted expression
3) A, where:
        A is the quoted expression
? 3
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair(A, nil), where:
        A is the quoted expression
1) [A], where:
        A is the quoted expression
2) Pair('quote', Pair(A, nil)), where:
        A is the quoted expression
3) A, where:
```

```
        A is the quoted expression
? 1
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair(A, nil), where:
        A is the quoted expression
1) [A], where:
        A is the quoted expression
2) Pair('quote', Pair(A, nil)), where:
        A is the quoted expression
3) A, where:
        A is the quoted expression
? 0
-- OK! --


---------------------------------------------------------------------
Problem 7 > Suite 2 > Case 1
(cases remaining: 5)


scm> (quote hello)
? hello
-- OK! --

scm> 'hello
? hello
-- OK! --

scm> ''hello
Choose the number of the correct choice:
0) (hello)
1) (quote hello)
2) (quote (quote (hello)))
3) hello
? 1
-- OK! --

scm> (quote (1 2))
? (1 2)
-- OK! --

scm> '(1 2)
? (1 2)
-- OK! --

scm> (quote (1 . 2))
? (1 . 2)
-- OK! --

scm> '(1 . (2))
? (1. (2))
-- Not quite. Try again! --

? (1. 2)
-- Not quite. Try again! --
```

```
? (1 . (2))
-- Not quite. Try again! --

? (1 . 2)
-- Not quite. Try again! --

? (1 2)
-- OK! --

scm> (car '(1 2 3))
? (1)
-- OK! --

scm> (cdr '(1 2))
? (2)
-- OK! --

scm> (car (car '((1))))
? SchemeError
-- Not quite. Try again! --

? (1)
-- OK! --

scm> (quote 3)
? 3
-- OK! --

scm> (eval (cons 'car '('(4 2))))
? 4
-- OK! --


---------------------------------------------------------------------
Problem 7 > Suite 3 > Case 1
(cases remaining: 4)

>>> from scheme_reader import *
>>> read_line(" (quote x) ")
Choose the number of the correct choice:
0) Pair('x', nil)
1) 'x'
2) Pair('quote', 'x')
3) Pair('quote', Pair('x', nil))
? 2
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair('x', nil)
1) 'x'
2) Pair('quote', 'x')
3) Pair('quote', Pair('x', nil))
? 3
-- OK! --

>>> read_line(" 'x ")
Choose the number of the correct choice:
0) Pair('x', nil)
```

```
1) Pair('quote', 'x')
2) 'x'
3) Pair('quote', Pair('x', nil))
? 3
-- OK! --

>>> read_line(" (a b) ")
Choose the number of the correct choice:
0) Pair('quote', Pair('a', 'b'))
1) Pair('quote', Pair('a', Pair('b', nil)))
2) Pair('quote', Pair(Pair('a', Pair('b', nil)), nil))
3) Pair('a', Pair('b', nil))
? 3
-- OK! --

>>> read_line(" '(a b) ")
Choose the number of the correct choice:
0) Pair('a', Pair('b', nil))
1) Pair('quote', Pair(Pair('a', Pair('b', nil)), nil))
2) Pair('quote', Pair('a', 'b'))
3) Pair('quote', Pair('a', Pair('b', nil)))
? 2
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair('a', Pair('b', nil))
1) Pair('quote', Pair(Pair('a', Pair('b', nil)), nil))
2) Pair('quote', Pair('a', 'b'))
3) Pair('quote', Pair('a', Pair('b', nil)))
? 3
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair('a', Pair('b', nil))
1) Pair('quote', Pair(Pair('a', Pair('b', nil)), nil))
2) Pair('quote', Pair('a', 'b'))
3) Pair('quote', Pair('a', Pair('b', nil)))
? 0
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair('a', Pair('b', nil))
1) Pair('quote', Pair(Pair('a', Pair('b', nil)), nil))
2) Pair('quote', Pair('a', 'b'))
3) Pair('quote', Pair('a', Pair('b', nil)))
? 1
-- OK! --

>>> read_line(" '((a)) ")
Choose the number of the correct choice:
0) Pair('quote', Pair(Pair('a'), nil))
1) Pair('quote', Pair(Pair(Pair('a', nil), nil), nil))
2) Pair('quote', Pair(Pair('a', nil), nil))
3) Pair('quote', Pair(Pair('a'), nil), nil)
4) Pair('quote', Pair(Pair('a', nil), nil), nil)
? 0
-- Not quite. Try again! --
```

```
Choose the number of the correct choice:
0) Pair('quote', Pair(Pair('a'), nil))
1) Pair('quote', Pair(Pair(Pair('a', nil), nil), nil))
2) Pair('quote', Pair(Pair('a', nil), nil))
3) Pair('quote', Pair(Pair('a'), nil), nil)
4) Pair('quote', Pair(Pair('a', nil), nil), nil)
? 3
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair('quote', Pair(Pair('a'), nil))
1) Pair('quote', Pair(Pair(Pair('a', nil), nil), nil))
2) Pair('quote', Pair(Pair('a', nil), nil))
3) Pair('quote', Pair(Pair('a'), nil), nil)
4) Pair('quote', Pair(Pair('a', nil), nil), nil)
? 4
-- Not quite. Try again! --

Choose the number of the correct choice:
0) Pair('quote', Pair(Pair('a'), nil))
1) Pair('quote', Pair(Pair(Pair('a', nil), nil), nil))
2) Pair('quote', Pair(Pair('a', nil), nil))
3) Pair('quote', Pair(Pair('a'), nil), nil)
4) Pair('quote', Pair(Pair('a', nil), nil), nil)
? 1
-- OK! --

_____
Problem 7 > Suite 3 > Case 2
(cases remaining: 3)

-- Already unlocked --

_____
Problem 7 > Suite 3 > Case 3
(cases remaining: 2)

-- Already unlocked --

_____
Problem 7 > Suite 3 > Case 4
(cases remaining: 1)

-- Already unlocked --

_____
OK! All cases for Problem 7 unlocked.

Backup... 100% complete
Backup successful for user: joseph.hayes@berkeley.edu

Checking for software updates...
OK is up to date
Josephs-Air-2:scheme JosephHayes$
```