Kuncheng Feng
CSC 416

AI Assignment: Basic List Processing

This assignment requires two replications from class exercise, and a session according to specifications

Task 1: Mimic “Lisp Session: CAR, CDR and CONS”

Command Prompt

```
Type :h and hit Enter for context help.

[1]> (car '(blue red yellow))
BLUE
[2]> (cdr '(blue red yellow))
(RED YELLOW)
[3]> (car '((1 2) buckle my shoe))
(1 2)
[4]> (cdr '((1 2) buckle my shoe))
(BUCKLE MY SHOE)
[5]> (car '("sunshine"))
"sunshine"
[6]> (cdr '("sunshine"))
NIL
[7]> (CONS 'ESPRESSO '(LATTE CAPPUCCINO))
(ESPRESSO LATTE CAPPUCCINO)
[8]> (CONS '(A B C) '(1 2 3))
((A B C) 1 2 3)
[9]> (CONS 'SYMBOL '())
(SYMBOL)
[10]> (bye)
Bye.

C:\Users\Kevin>
```

Task 2: Mimic “Redacted Lisp Session: Three additional referencers and constructors”

Command Prompt

```
[1]> (setf oo-languages '(simula smalltalk java clos))
(SIMULA SMALLTALK JAVA CLOS)
[2]> oo-languages
(SIMULA SMALLTALK JAVA CLOS)
[3]> 'oo-languages
OO-LANGUAGES
[4]> (quote oo-languages)
OO-LANGUAGES
[5]> (car oo-languages)
SIMULA
[6]> (cdr 00-languages)

*** - SYSTEM::READ-EVAL-PRINT: variable 00-LANGUAGES has no value
The following restarts are available:
USE-VALUE      :R1      Input a value to be used instead of 00-LANGUAGES.
STORE-VALUE    :R2      Input a new value for 00-LANGUAGES.
ABORT          :R3      Abort main loop
Break 1 [7]> :a
[8]> (cdr oo-languages)
(SMALLTALK JAVA CLOS)
[9]> (car (cdr oo-languages))
SMALLTALK
[10]> (cdr (cdr oo-languages))
(JAVA CLOS)
```

Command Prompt

```
[11]> (cadr oo-languages)
SMALLTALK
[12]> (cddr oo-languages)
(JAVA CLOS)
[13]> (first oo-languages)
SIMULA
[14]> (second oo-languages)
SMALLTALK
[15]> (third oo-languages)
JAVA
[16]> (nth 2 oo-languages)
JAVA
[17]> (setf numbers '(1 2 3))
(1 2 3)
[18]> (setf letters '(a b c))
(A B C)
[19]> (cons numbers letters)
((1 2 3) A B C)
[20]> (list numbers letters)
((1 2 3) (A B C))
[21]> (append numbers letters)
(1 2 3 A B C)
[22]> (list numbers (cdr numbers) (cddr numbers))
((1 2 3) (2 3) (3))
[23]> (append numbers (cdr numbers) (cddr numbers))
```

Command Prompt

```
((1 2 3) (2 3) (3))
[23]> (append numbers (cdr numbers) (cddr numbers))
(1 2 3 2 3 3)
[24]> (setf elle '(ant bat cat dog eel))
(ANT BAT CAT DOG EEL)
[25]> (car (cdr (cdr (cdr elle))))
DOG
[26]> (nth 3 elle)
DOG
[27]> (setf a 'apple b 'peach c 'cherry)
CHERRY
[28]> (cons a (cons b (cons c ())))
(APPLE PEACH CHERRY)
[29]> (list a b c)
(APPLE PEACH CHERRY)
[30]> (setf x '(red blue) y '(green yellow))
(GREEN YELLOW)
[31]> (cons (car x) (cons (car (cdr x)) y))
(RED BLUE GREEN YELLOW)
[32]> (append x y)
(RED BLUE GREEN YELLOW)
[33]> (bye)
Bye.

C:\Users\Kevin>
```

Task 3: Create a Lisp session according to specification

```
[1]> (setf english '(one two three four))
(ONE TWO THREE FOUR)
[2]> (setf french '(un deux trois quatre))
(UN DEUX TROIS QUATRE)
[3]> (setf pair1 (list (car english) (car french)))
(ONE UN)
[4]> (setf pair2 (list (car (cdr english)) (car (cdr french))))
(TWO DEUX)
[5]> (setf pair3 (list (nth 2 english) (nth 2 french)))
(THREE TROIS)
[6]> (setf pair4 (list (nth 3 english) (nth 3 french)))
(FOUR QUATRE)
[7]> (setf dictionary (list pair1 pair2 pair3 pair4))
((ONE UN) (TWO DEUX) (THREE TROIS) (FOUR QUATRE))
[8]> (setf ef-words (append pair1 pair2 pair3 pair4))
(ONE UN TWO DEUX THREE TROIS FOUR QUATRE)
[9]> (setf alt-words (append english french))
(ONE TWO THREE FOUR UN DEUX TROIS QUATRE)
```

1. Bind the symbol ENGLISH to the list (ONE TWO THREE FOUR).

```
(setf english '(one two three four))
```

2. Bind the symbol FRENCH to the list (UN DEUX TROIS QUATRE).

```
(setf french '(un deux trois quatre))
```

3. Bind the symbol PAIR1 to the pair (ONE UN) by means of a form involving one occurrence of LIST, two occurrences of CAR, and the symbols ENGLISH and FRENCH.

```
(setf pair1 (list (car english) (car french)))
```

4. Bind the symbol PAIR2 to the pair (TWO DEUX) by means of a form involving one occurrence of LIST, two occurrences of CAR, two occurrences of CDR, and the symbols ENGLISH and FRENCH.

```
(setf pair2 (list (car (cdr english)) (car (cdr french))))
```

5. Bind the symbol PAIR3 to the pair (THREE TROIS) by means of a form involving one occurrence of LIST, two occurrences of NTH, and the symbols ENGLISH and FRENCH.

```
(setf pair3 (list (nth 2 english) (nth 2 french)))
```

6. Bind the symbol PAIR4 to the pair (FOUR QUATRE) by means of a form involving one occurrence of LIST, two occurrences of NTH, and the symbols ENGLISH and FRENCH.

```
(setf pair4 (list (nth 3 english) (nth 3 french)))
```

7. Bind the symbol DICTIONARY to the list ((ONE UN) (TWO DEUX) (THREE TROIS) (FOUR QUATRE)) by means of a form involving one occurrence of LIST, and the symbols PAIR1, PAIR2, PAIR3 and PAIR4.

```
(setf dictionary (list pair1 pair2 pair3 pair4))
```

8. Bind the symbol EF-WORDS to the list (ONE UN TWO DEUX THREE TROIS FOUR QUATRE) by means of a form involving one occurrence of APPEND, and the symbols PAIR1, PAIR2, PAIR3 and PAIR4.

```
(setf ef-words (append pair1 pair2 pair3 pair4))
```

9. Bind the symbol ALT-WORDS to the list (ONE TWO THREE FOUR UN DEUX TROIS QUATRE) by means of a form involving one occurrence of APPEND, and the symbols ENGLISH and FRENCH.

```
(setf alt-words (append english french))
```