(CSCI 421 VA) Project Four
Define Type and Write Functions to Handle the Type

Objectives:

- Write data constructor with parameters
- Write type constructor with parameters
- Write recursively defined type constructors

Problem Description:

The student shall turn in definition of mylist and three functions to handle mylist. In YourNameProjFour.sml file, the student needs to write the following type and functions

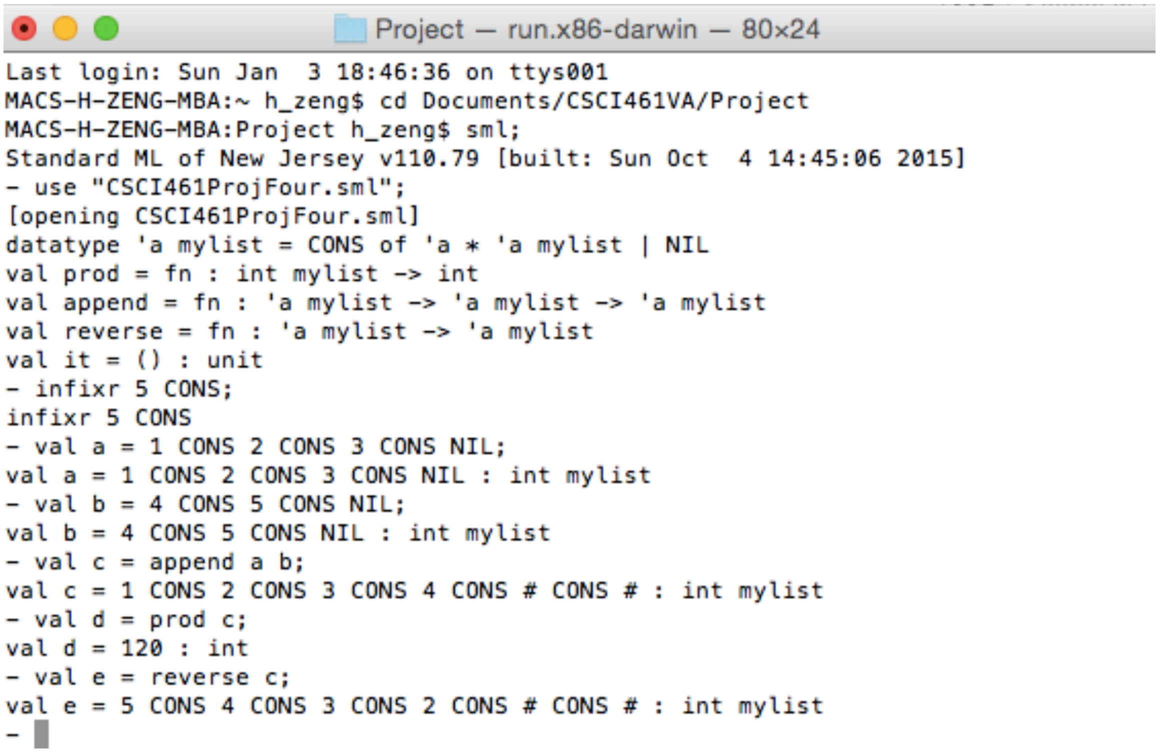
1. Define mylist type that is the same as the one on page 172 of the textbook.
2. Define a function prod that returns the product of all integers in a mylist of integers. (refer to problem Exercise 6 on page 178)
3. Write a function append of type ‘a mylist -> ‘a mylist -> ‘a mylist that takes two mylist values, a and b, and return a mylist contains all the values in a followed by all the values in b. (refer to Exercise 8 on page 178)
4. **Write** a function reverse of type ‘a list -> ‘a list that takes a mylist x and return a mylist of all elements of x, in a reverse order. (refer to Exercise 7on page 178)

Sample Run:

Please see the screen shot on next page. Please notice how infixr 5 is used. (Refer to examples on page 174). Also please notice that the result list may not fully displayed. For instance, for list with element 1, 2, 3, 4, 5, it may only display as 1 CONS 2 CONS 3 CONS # CONS #

Due Date:

The file needs to be submitted through blackboard link on time. The file name should be YourNameProjFour.sml. Due date will be announced on blackboard.



```
Last login: Sun Jan  3 18:46:36 on ttys001
MACS-H-ZENG-MBA:~ h_zeng$ cd Documents/CSCI461VA/Project
MACS-H-ZENG-MBA:Project h_zeng$ sml;
Standard ML of New Jersey v110.79 [built: Sun Oct  4 14:45:06 2015]
- use "CSCI461ProjFour.sml";
[opening CSCI461ProjFour.sml]
datatype 'a mylist = CONS of 'a * 'a mylist | NIL
val prod = fn : int mylist -> int
val append = fn : 'a mylist -> 'a mylist -> 'a mylist
val reverse = fn : 'a mylist -> 'a mylist
val it = () : unit
- infixr 5 CONS;
infixr 5 CONS
- val a = 1 CONS 2 CONS 3 CONS NIL;
val a = 1 CONS 2 CONS 3 CONS NIL : int mylist
- val b = 4 CONS 5 CONS NIL;
val b = 4 CONS 5 CONS NIL : int mylist
- val c = append a b;
val c = 1 CONS 2 CONS 3 CONS 4 CONS # CONS # : int mylist
- val d = prod c;
val d = 120 : int
- val e = reverse c;
val e = 5 CONS 4 CONS 3 CONS 2 CONS # CONS # : int mylist
- 
```