Problem Set 2

OUT: 13 SEPTEMBER, 1994
DUE: 22 SEPTEMBER, 1994

Goals

Introduction to Linked Lists.

Reading Assignment

Read chapters 11 and 12 in *Condensed Pascal.*

Programming Assignment

Write a program which keeps a linked list of the *unique* words in a text file, along with a count as to how many times each unique word appeared in the text. Remove all punctuation and ignore all capitalization. For instance, the text *Romeo, Romeo! wherefore art thou Romeo?* has 4 unique words, *romeo* with a count of 3, *wherefore, art* and *thou* each with a count of 1.

The output should be the total number of words, the number of unique words, and a list of the unique words with their count. To make sure your code really works, you should take a small file and watch what it does with each word read. Also test your program on the long file which will be copied to your `INBOX`.

⋆ *Be sure to write Loop Invariants for each iteration block and to record the input and output of your small test run.*

Use your MyString subroutines to copy and compare MyStrings, as well as to input the words. You will need an additional routine to change a MyString into all lower-case letters. Then define a data object, a list of strings,

```
type  MyList = ^ MyListElement ;
      MyListCoupon = MyList ;
      MyListElement = record   theString : MyString ;
                               count : integer ;
                               next : MyList ;   end ;
```

Now write the following list of string manipulation routines,

```
function CreateMyList() : MyList ;
{ returns an empty MyList }

function IsEmptyMyList( ml : MyList ) : boolean ;
{ returns true if ml is empty, false otherwise. }

procedure PrintMyList( ml : MyList ) ;
{ prints out ml }

procedure AddToFrontMyList( var ml : MyList;
    s : MyString; count : integer ) ;
{ adds MyString element to head of MyList and initializes count }

function FindMyList( ml : MyList; s : MyString ) : MyListCoupon ;
{ finds MyString s on MyList ml. If found, returns a coupon
  to the found MyList element, else returns NIL }

procedure IncrCountMyList( var ml : MyList; coupon : MyListCoupon ) ;
{ increments the count of MyList element indicated by coupon.
  Coupon was returned by a call to FindMyList. }

function LengthMyList( ml : MyList ) : integer ;
{ returns the length of MyList, i.e. number of elements. }

function SumCountMyList( ml : MyList ) : integer ;
{ returns the sum of the count fields over all elements. }
```

Sample Run

The input *Romeo, Romeo! wherefore art thou Romeo?* gives output,

```
6 words in the file,
4 unique words. They are:
   thou       1
   art        1
   wherefore  1
   romeo      3
```