Pascal Examples

OUT: 2 SEPTEMBER, 1994

```
--------------------------- example5mod.pas --------------------

[ENVIRONMENT]
module MyString1(input,output) ;

(* example program 5: strings.
   Data Objects
   (c) Burt Rosenberg, Univ. of Miami 1994
*)

const
  MYSTRINGLEN = 30 ;
  MYEOF = -1 ;

type
  MyString = record
             length : integer ;
             data : array [1..MYSTRINGLEN+1] of char ;
             end ;

procedure writeln_MyString( ms : myString ) ;
(*
   Writes the string ms, followed by a newline.
*)
var i : integer ;
begin
   i := 0 ;
   while ( i<ms.length ) do begin
      i := i + 1 ;
      write(ms.data[i]) ;
   end ;
   writeln ;
end ;

function readln_MyString( var ms : myString ) : integer ;
(*
   Reads the string into ms, up to a new line.
   Handles case of line longer than maximum allowed, but does not
      report an error.
   Returns:
```

```
      the length of the string, 0 for an empty line
      MYEOF, and an empty string, if end of file encountered.
*)
begin
   ms.length := 0 ;
   if ( not eof ) then begin
      while ( not eoln ) do begin
         if ( ms.length < MYSTRINGLEN ) then
            ms.length := ms.length + 1 ;
         read(ms.data[ms.length]) ;
      end ;
      readln ;
      readln_MyString := ms.length ;
   end else readln_MyString := MYEOF ;
end ;

end.

-------------------- example6 ----------------------

[INHERIT('EXAMPLE5MOD')]
program compare( input, output ) ;

(* example 6, comparing two MyStrings,
   version 1: a very clear style.
   (c) Burton Rosenberg, U. of Miami 1994
*)


function compare_MyString( s : MyString ; t : MyString ) : boolean ;
var
   i : integer ;
   flag : boolean ;
begin
   if ( s.length <> t.length ) then
      compare_MyString := false
   else begin
      (* precondition, s.length = t.length *)
      flag := true ;
      i := 1 ;
      (* L.I. for all (0<j<i) s.data[j] = t.data[j] *)
      while (flag) do
         if (i>s.length) then flag := false
         else if (s.data[i]=t.data[i]) then i := i + 1
```

```
            else flag := false ;
      compare_MyString := i > s.length
   end (*else*)
end ;

var
   a, b : MyString  ;
   i : integer ;

begin
   writeln('Enter string 1: ') ;
   i := readln_MyString( a ) ;
   writeln('Enter string 2: ') ;
   i := readln_MyString( b ) ;
   if ( compare_MyString( a, b ) ) then
      writeln('The strings are equal')
   else writeln('The strings are NOT equal') ;
end .

-------------------- example7 ----------------------

[INHERIT('EXAMPLE5MOD')]
program compare( input, output ) ;

(* example 7, comparing two MyStrings,
   version 2: using Sentinels.
   (c) Burton Rosenberg, U. of Miami 1994
*)


function compare_MyString( s : MyString ; t : MyString ) : boolean ;
(*
   returns true if mystring s equals mystring t, false elsewise.
   requires that s.data allows index MYSTRINGLEN+1, for sentinel value.
*)
var
   i : integer ;
begin
   if ( s.length <> t.length ) then
      compare_MyString := false
   else begin
      (* precondition, s.length = t.length *)
      i := s.length + 1 ;
      s.data[i] := '.' ;
```

```
      t.data[i] := ',' ;
      i := 1 ;
      (* L.I. for all (0<j<i) s.data[j] = t.data[j] *)
      while (s.data[i]=t.data[i]) do i := i + 1  ;
      compare_MyString := i > s.length
   end (*else*)
end ;

var
   a, b : MyString  ;
   i : integer ;

begin
   writeln('Enter string 1: ') ;
   i := readln_MyString( a ) ;
   writeln('Enter string 2: ') ;
   i := readln_MyString( b ) ;
   if ( compare_MyString( a, b ) ) then
      writeln('The strings are equal')
   else writeln('The strings are NOT equal') ;
end .
```