Example 1

OUT: 30 AUGUST, 1994

```
program max(input,output) ;

(* example 1 program: find the max in an array.
   Loop Invariants
   (c) Burton Rosenberg, Univ. of Miami, 1994
*)

const
  N = 10 ;
type
  myarray = array [0..N] of integer ;
var
  i : integer ;
  imax : integer ;
  a : myarray ;

begin

  writeln('Find the maximum number in an array.') ;
  writeln('Enter ', N:0, ' integers:') ;
  for i := 1 to N do read(A[i]) ;

  (* Precondition N > 0 *)

  imax := 1 ;
  i := 2 ;

  (* Loop Invariant: A[imax] = max(0<j<i)A[j] *)

  while ( i<=N ) do begin
    if (A[i]>A[imax]) then imax := i ;
    i := i + 1 ;
  end ;

  (* LI + Termination = Goal, i.e.
     Postcondition: A[imax] is max. in array *)

  writeln('The maximum is ',A[imax]:0) ;
end.
```

Example 2

```
program max(input,output) ;

(* example 2 program: sum an array.
   Loop Invariants
   (c) Burton Rosenberg, Univ. of Miami, 1994
*)

const
  N = 10 ;
type
  myarray = array [0..N] of integer ;
var
  i : integer ;
  sum : integer ;
  a : myarray ;

begin

  writeln('Sum the numbers in an array.') ;
  writeln('Enter ', N:0, ' integers:') ;
  for i := 1 to N do read(A[i]) ;

  (* Precondition N > 0 *)

  sum := 0 ;
  i := 1 ;

  (* Loop Invariant: sum = Sum(0<j<i)A[j] *)

  while ( i<=N ) do begin
    sum := sum + A[i] ;
    i := i + 1 ;
  end ;

  (* LI + Termination = Goal, i.e.
     Postcondition: sum is sum of array *)

  writeln('The sum is ',sum:0) ;
end.
```

Example 3

```
program zero(input,output) ;

(* example 3 program: find a zero entry in an array.
   Loop Invariants
   (c) Burton Rosenberg, Univ. of Miami, 1994
*)

const
  N = 10 ;
  FINDNUM = 0 ;

type
  myarray = array [0..N] of integer ;
var
  i : integer ;
  a : myarray ;

begin

  writeln('Find a ', FINDNUM:0, ' in an array.') ;
  writeln('Enter ', N:0, ' integers:') ;
  for i := 1 to N do read(A[i]) ;

  (* Precondition N > 0 *)
  i := 1 ;

  (* Loop Invariant: for all (0<j<i), A[j] <> FINDNUM *)

  while ( (A[i]<>FINDNUM) AND (i<N) ) do
    i := i + 1 ;

  (* LI + Termination = Goal, i.e.
     Postcondition: either A[i]=FINDNUM, or FINDNUM not in array *)

  write(FINDNUM:0) ;
  if (A[i]<>FINDNUM) then write(' is not')
  else write(' is element ', i:0) ;
  writeln(' in the array.') ;

end.
```

Example 4

```
program findnumSentinel(input,output) ;

(* example 4 program: find a number in an array.
   Loop Invariants
   and Sentinel values.
   (c) Burton Rosenberg, Univ. of Miami, 1994
*)

const
  N = 10 ;
  FINDNUM = 0 ;
type
  myarray = array [0..N] of integer ;
var
  i : integer ;
  a : myarray ;

begin

  writeln('Find a ', FINDNUM:0, ' in an array.') ;
  writeln('Enter ', N:0, ' integers:') ;
  for i := 1 to N do read(A[i]) ;

  (* Precondition N > 0 *)
  i := N ;
  A[0] := FINDNUM ;

  (* Loop Invariant: for all (i<j<N), A[j] <> FINDNUM *)

  while ( A[i]<>FINDNUM ) do
    i := i - 1 ;

  (* LI + Termination = Goal, i.e.
     Postcondition: i>0 implies A[i]=FINDNUM,
        else FINDNUM not in array *)

  write(FINDNUM:0) ;
  if ( i = 0 ) then write(' is not')
  else write(' is element ', i:0) ;
  writeln(' in the array.') ;

end.
```

Example 5

```
program myString(input,output) ;

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

const
  STRINGN = 10 ;
  MYEOF = -1 ;

type
  myString = record
             length : integer ;
             data : array [1..STRINGN] of char ;
             end ;
var
   ms : mystring ;

procedure writeln_String( 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_String( 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 if end of file encountered.
```

```
*)
begin
   ms.length := 0 ;
   if ( not eof ) then begin
      while ( not eoln ) do begin
         if ( ms.length < STRINGN ) then
            ms.length := ms.length + 1 ;
         read(ms.data[ms.length]) ;
      end ;
      readln ;
      readln_String := ms.length ;
   end else readln_String := MYEOF ;
end ;

(***** MAIN PROGRAM *******)
begin
   writeln('Test program for string objects.') ;
   writeln('Enter lines, ', STRINGN:0,' characters maximum on a line.') ;
   writeln('^Z [VAX eof] to terminate.') ;
   while (readln_String( ms ) <> MYEOF ) do writeln_String( ms ) ;
end.
```