```
-- Ejercicios hechos en clase sobre listas estáticas
--  14 Noviembre 2014
--  Marisa Navarro

WITH Ada.Integer_Text_IO, Ada.Text_IO;
USE Ada.Integer_Text_IO, Ada.Text_IO;

PROCEDURE EjerciciosListas IS

   -- Declaraciones de constantes, tipos, subprogramas y variables

   Max : CONSTANT Positive := 50;

   TYPE T_Numeros IS ARRAY (1 .. Max) OF Integer;

   TYPE T_Lista IS
      RECORD
         Numeros : T_Numeros;
         Cont    : Natural;
      END RECORD;

   ------------------------------------------------------------
   PROCEDURE Leer_Lista (L: OUT T_Lista) IS
    -- entrada: secuencia S de enteros terminada en 0  (EE)
    -- pre: S contiene como mucho Max números.
    -- post: L contiene todos los elementos de la secuencia S.
      N : Natural;
   BEGIN
      L. Cont := 0;
      Get(N);
      WHILE N/=0 LOOP
         L.Numeros(L.Cont+1):=N;
         Get(N);
         L.Cont:= L.Cont+1;
      END LOOP;
   END Leer_Lista;
   ------------------------------------------------------------

   ------------------------------------------------------------
   PROCEDURE Escribir_Lista (L: IN T_Lista) IS
    -- salida: secuencia de enteros S   (SE)
    -- post: S es la secuencia de elementos de la lista L.
   BEGIN
      Put("(");
      FOR I IN 1.. L.Cont-1 LOOP
         Put(L.Numeros(I), 0);
         Put(", ");
      END LOOP;
      IF L.Cont>0 THEN
         Put(L.Numeros(L.Cont), 0);
      END IF;
      Put("). ");
   END Escribir_Lista;
   ------------------------------------------------------------

   ------------------------------------------------------------
```

```
   FUNCTION Esta (L: T_Lista; Num: Integer) RETURN Boolean IS
    -- post: devuelve true sii Num está en la lista L
      I : Natural := 1;
   BEGIN
      WHILE I<= L.Cont AND THEN L.Numeros(I)/= Num LOOP
         I:= I+1;
      END LOOP;
      RETURN ( I<=L.Cont );
   END Esta;
   ------------------------------------------------------------


   ---------------------------------------------------------------------------
   FUNCTION Esta_En_Lista_Ord (L: T_Lista; Num: Integer) RETURN Boolean IS
    -- pre: L ordenada crecientemente
    -- post: devuelve true sii Num está en la lista L
      I : Natural := 1;
   BEGIN
      WHILE I<= L.Cont AND THEN L.Numeros(I)<Num LOOP
         I:= I+1;
      END LOOP;
      RETURN ( I<=L.Cont AND THEN L.Numeros(I)=Num );
   END Esta_En_Lista_Ord;
   ------------------------------------------------------------


   ---------------------------------------------------------------------------
   PROCEDURE Elimina_Primera_Aparicion (L: IN OUT T_Lista; Num: IN Integer) IS
   -- post: devuelve L sin la primera aparición de Num (si no hay, deja L igual)
      I : Natural := 1;
   BEGIN
      WHILE I<= L.Cont AND THEN L.Numeros(I)/=Num LOOP
         I:= I+1;
      END LOOP;
      -- Aqui o bien I es la posición donde está Num o bien I= L.Cont +1
      IF I <= L.Cont THEN
         L.Numeros(I..L.Cont-1) := L.Numeros(I+1.. L.Cont);
         L.Cont := L.Cont -1;
      END IF;
   END Elimina_Primera_Aparicion;
   ---------------------------------------------------------------------------


   ------------------------------------------------------------
   PROCEDURE Inversa_Lista (L: IN T_Lista; Lres: OUT T_Lista) IS
      -- post: Lres es la lista inversa de L;
      J : Natural := 0;
   BEGIN
      FOR I IN REVERSE 1.. L.Cont LOOP
         Lres.Numeros(J+1):= L.Numeros(I);
         J:= J+1;
      END LOOP;
      Lres.Cont := J;
   END Inversa_Lista;
   ------------------------------------------------------------

   Lis1, Lis2, Lis3 : T_Lista;

BEGIN
   -- Cuerpo del programa principal para probar los subprogramas
```

```
Lis1.Cont :=10;
Lis1.Numeros(1..10):= (1,2,3,4,5,6,7,8,9,10);

New_LINE; Put_Line("Mi lista Lis1 es ");
New_Line; Put("Lis1 = ");
Escribir_Lista(Lis1);
New_Line; New_Line;

Put_Line(" Probando Inversa_Lista de Lis1 y dejandola en Lis2 ");
New_Line;
Inversa_Lista(Lis1,Lis2);
Put("--->   Lis2 = ");
Escribir_Lista(Lis2);
New_Line; New_Line;

Put_Line(" Probando Esta_En_Lista_Ord con Lis1 y el numero 5 ");
New_Line;
IF Esta_En_Lista_Ord(Lis1,5) THEN Put("---> SI");
ELSE Put("---> NO");
END IF;
New_Line; New_Line;

Put_Line(" Probando Esta_En_Lista_Ord con Lis1 y el numero 12 ");
New_Line;
IF Esta_En_Lista_Ord(Lis1,12) THEN Put("---> SI");
ELSE Put("---> NO");
END IF;
New_Line; New_Line;

Put_Line(" Probando Esta con Lis2 y el numero 5 ");
New_Line;
IF Esta(Lis2,5) THEN Put("---> SI");
ELSE Put("---> NO");
END IF;
New_Line; New_Line;

Put_Line(" Probando Esta con Lis2 y el numero 12 ");
New_Line;
IF Esta_En_Lista_Ord(Lis2,12) THEN Put("---> SI");
ELSE Put("---> NO");
END IF;
New_Line; New_Line;

Put_Line(" Probando Eliminar la primera aparicion del 5 en Lis2 nos queda");
New_Line;
Elimina_Primera_Aparicion(Lis2,5);
Put("---> Lis2 = ");
Escribir_Lista(Lis2);
New_Line; New_Line;

Put_Line(" Ahora eliminamos la primera aparicion del 10 en Lis2 y nos queda");
New_Line;
Elimina_Primera_Aparicion(Lis2,10);
Put("---> Lis2 = ");
Escribir_Lista(Lis2);
New_Line; New_Line;
```

```
   Put_Line(" Ahora eliminamos la primera aparicion del 1 en Lis2 y nos queda");
   New_Line;
   Elimina_Primera_Aparicion (Lis2,1);
   Put("---> Lis2 = ");
   Escribir_Lista(Lis2);
   New_Line; New_Line;

   Put_Line(" Probando Leer_Lista y dejandola en Lis3 ");
   New_Line;
   Put_Line(" Dame una serie de numeros (el cero es el centinela y puedes dar como maximo 50
   numeros) ");
   New_Line;
   Leer_Lista(Lis3);
   New_Line;
   Put("--->   Lis3 = ");
   Escribir_Lista(Lis3);
   New_Line;
   Put("--->   Lis3 tiene "); Put(Lis3.Cont, 0); Put_Line(" numeros");
   New_Line;

   Put_Line("Fin de las pruebas");

   Skip_Line; Skip_Line;  -- (para ejecutarlo fuera de AdaGIDE sin que se cierre)

END EjerciciosListas;
```