C9-1

Algorithm

Package Specification

```
1. Declare the type.
2. Create a new package within the package specification to provide predefined
   functions for the enumeration type.
3. Declare the function prototypes for both the predecessor and successor functions.
```

Package Implementation

```
1. Successor function:
    a. if the input_value = Type'Last then
           return Type'First
    b. else
           return Type'Succ(input_value);

2. Predecessor function:
    a. if the input_value = Type'First then
           return Type'Last
    b. else
           return Type'Pred(input_value);
```

Code Listing

```
GNAT 3.13p  (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Checking: c:/docume~2/jk/desktop/16070/codeso~1/my_type_package.ads (source file time stamp: 2003-09-24 01:59:20)
```

```
---------------------------------------------------------
-- Package specified to declare the type and two functions
-- Specifier : Jayakanth Srinivasan
-- Date Last Modified : 09/23/2003
---------------------------------------------------------

with Ada.Text_Io;

package My_Type_Package is

   type Day is
      (Monday,
       Tuesday,
       Wednesday,
       Thursday,
       Friday,
       Saturday,
       Sunday);
```

```
  19.    package Day_Io is new Ada.Text_Io.Enumeration_Io(Enum => Day);
  20.
  21.    function Successor (
  22.          Day_In : Day )
  23.      return Day;
  24.
  25.    function Predecessor (
  26.          Day_In : Day )
  27.      return Day;
  28. end My_Type_Package;

 28 lines: No errors
```

```
GNAT 3.13p  (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/jk/desktop/16070/codeso~1/my_type_package.adb (source file time stamp: 2003-
09-24 02:00:04)

     1. ---------------------------------------------------------
     2. -- Package implementation of My_type package
     3. -- Implementer : Jayakanth Srinivasan
     4. -- Date Last Modified : 09/23/2003
     5. ---------------------------------------------------------
     6.
     7. package body My_Type_Package is
     8.
     9.
    10.    function Successor (
    11.          Day_In : Day )
    12.      return Day is
    13.    begin
    14.       if Day_In = Day'Last then
    15.          return Day'First;
    16.       else
    17.          return Day'Succ(Day_In);
    18.       end if;
    19.    end Successor;
    20.
    21.
    22.    function Predecessor (
    23.          Day_In : Day )
    24.      return Day is
    25.    begin
    26.       if Day_In = Day'First then
    27.          return Day'Last;
    28.       else
    29.          return Day'Pred(Day_In);
    30.       end if;
    31.    end Predecessor;
    32. end My_Type_Package;

 32 lines: No errors
```

C9-2

```
GNAT 3.13p  (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/jk/desktop/16070/codeso~1/test_types.adb (source file time stamp: 2003-09-24
02:03:20)

    1. ----------------------------------------------------------
    2. -- Program to test the package implementation
    3. -- Programmer : Jayakanth Srinivasan
    4. -- Date Last Modified : 09/23/2003
    5. ----------------------------------------------------------
    6. with My_Type_Package;
    7. with Ada.Text_Io;
    8.
    9. procedure Test_Types is
   10.    My_Day : My_Type_Package.Day;
   11.
   12. begin
   13.    -- initialize the day to monday
   14.    My_Day := My_Type_Package.Monday;
   15.
   16.    My_Type_Package.Day_Io.Put(My_Type_Package.Successor(My_Day));
   17.    Ada.Text_Io.New_Line;
   18.
   19.    My_Type_Package.Day_Io.Put(My_Type_Package.Predecessor(My_Day));
   20.    Ada.Text_Io.New_Line;
   21.
   22.    -- change the day to sunday
   23.    My_Day := My_Type_Package.Sunday;
   24.
   25.    My_Type_Package.Day_Io.Put(My_Type_Package.Successor(My_Day));
   26.    Ada.Text_Io.New_Line;
   27.
   28.    My_Type_Package.Day_Io.Put(My_Type_Package.Predecessor(My_Day));
   29.    Ada.Text_Io.New_Line;
   30. end Test_Types;
   31.
   32.

 32 lines: No errors
```