C-16 Solutions

1. Record Declaration

```
type Aircraft_Information is record
  Aircraft_Number : Integer;
  Latitude : Float;
  Longitude : Float;
  Heading : Float;
  Velocity : Float;
end record;
```

Note there are multiple ways to do this record declaration. You may for instance choose to use a hierachical record wherein the position information (Latitude, Longitude, Heading, Velocity) is a record within Aircraft_Information as shown below:

```
type Position_Information is record
  Latitude : Float;
  Longitude : Float;
  Heading : Float;
  Velocity : Float;
end record;

type Aircraft_Information is record
  Aircraft_Number : Integer;
  Aircraft_Position : Position_Information;
end record;
```

2. Ada Program

Data Structures

Array of 10 elements of Type Aircraft_Information

Subprograms

- Function to create the array of aircraft
- Procedure to sort the contents of the array based on latitude
- Procedure to compute and display the distances between the first aircraft and all other aircraft.

Algorithms

Create_Aircraft:
    For I in 1 .. 10
        Prompt the user to input relevant information
        Store the information in Array(I)
    Return Array to the main program

Sort_Aircraft:

    For I in 1 .. Num_of_Aircraft -1
        For J in I+1 .. Num_Of_Aircraft
            If Array(I).Latitude > Array(J).Latitude
                Swap the records in Array(I) and Array(J)
    Return sorted array to the user

Compute_Distances:

    For I in 2 .. Num_Of_Aircraft
        Compute difference in latitudes (dlat)
        Compute the differences in longitude (dlon)
        Covert the differences into distances using the WGS-84
        approximations in the handout (dlat_dist, dlon_dist)
        Distance between the aircraft = sqrt(dlat_dist^2 + dlon_dist^2)
        Display computed distance to the user.

Main Program:
    Create aircraft using the Create_Aircraft function
    Sort the aircraft
    Compute the distances and display it to the user

Code Listing

My_Aircraft Package Specification

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

Checking: c:/docume~2/joeb/desktop/16070/codeso~1/my_aircraft.ads (source file time stamp: 2003-10-08
18:10:40)

 1.
 2. ------------------------------------------------
 3. -- Package to specify aircraft parameters and the
 4. -- related subprograms
 5. -- Specifier : Joe B
 6. -- Date Last Modified : 10/07/03
 7. ------------------------------------------------
 8. package My_Aircraft is
 9.    Num_of_Aircraft : constant Integer := 10;
10.
11.    type Aircraft_Information is
```

```
12.      record
13.         Aircraft_Number : Integer;
14.         Latitude        : Float;
15.         Longitude       : Float;
16.         Heading         : Float;
17.         Velocity        : Float;
18.      end record;
19.
20.   type Aircraft_Array is array (1 .. Num_Of_Aircraft) of Aircraft_Information;
21.
22.   Latitude_Conversion : constant Float := 1852.24;
23.   Longitude_Conversion : constant Float := 1314.13 ;
24.
25.   function Get_Aircraft_Info return Aircraft_Array;
26.
27.   procedure Sort_Aircraft (
28.         Input_Array : in out Aircraft_Array );
29.
30.   procedure Compute_Distances (
31.         Input_Array : in     Aircraft_Array );
32. end My_Aircraft;
```

32 lines: No errors

My_Aircraft Package Body

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

Compiling: c:/docume~2/joeb/desktop/16070/codeso~1/my_aircraft.adb (source file time stamp: 2003-10-08 18:26:26)

```
1.
2. ----------------------------------------------
3. -- Package body for My_Aircraft
4. -- Programmer: Joe B
5. -- Date Last Modified : 10/07/03
6. ----------------------------------------------
7. with Ada.Text_Io;
8. with Ada.Integer_Text_Io;
9. with Ada.Float_Text_Io;
10. with Ada.Numerics.Elementary_Functions;
11.
12. package body my_aircraft is
13.
14.
15.   function Get_Aircraft_Info return Aircraft_Array is
16.      Output_Array : Aircraft_Array;
17.   begin
18.      for I in 1 .. Num_of_Aircraft loop
19.         Ada.Text_Io.Put("Please Enter Information of Aircraft");
20.         Ada.Text_Io.Put(Integer'Image(I));
21.         Ada.Text_Io.New_Line;
22.
23.         Ada.Text_Io.Put("Aircraft Id : ");
24.         Ada.Integer_Text_IO.Get(Output_Array(I).Aircraft_Number);
25.         Ada.Text_Io.Skip_Line;
26.
```

```
27.       Ada.Text_Io.Put("Latitude : ");
28.       Ada.Float_Text_Io.Get(Output_Array(I).Latitude);
29.       Ada.Text_Io.Skip_Line;
30.
31.       Ada.Text_Io.Put("Longitude: ");
32.       Ada.Float_Text_Io.Get(Output_Array(I).Longitude);
33.       Ada.Text_Io.Skip_Line;
34.
35.
36.       Ada.Text_Io.Put("Heading : ");
37.       Ada.Float_Text_Io.Get(Output_Array(I).Heading);
38.       Ada.Text_Io.Skip_Line;
39.
40.       Ada.Text_Io.Put("Velocity: ");
41.       Ada.Float_Text_Io.Get(Output_Array(I).Velocity);
42.       Ada.Text_Io.Skip_Line;
43.     end loop;
44.     return Output_Array;
45.   end Get_Aircraft_Info;
46.
47.   procedure Sort_Aircraft ( Input_Array : in out Aircraft_Array) is
48.       Temp : Aircraft_Information;
49.   begin
50.     for I in 1 .. Num_of_Aircraft-1 loop
51.       for J in I+1 .. Num_Of_Aircraft loop
52.         if Input_Array(I).Latitude > Input_Array(J).Latitude then
53.           Temp := Input_Array(I);
54.           Input_Array(I) := Input_Array(J);
55.           Input_Array(J) := Temp;
56.         end if;
57.       end loop;
58.     end loop;
59.   end Sort_Aircraft;
60.
61.
62.   procedure Compute_Distances (Input_Array : in Aircraft_Array) is
63.     Distance, dlat, dlat_dist, dlon, dlon_dist : Float;
64.   begin
65.     for I in 2 .. Num_Of_Aircraft loop
66.       Dlat := Input_Array(1).Latitude - Input_Array(I).Latitude;
67.
68.       dlat_dist := dlat * 60.0 * latitude_conversion;
69.       dlat_dist := dlat_dist * dlat_dist;
70.
71.       Dlon := Input_Array(1).Longitude - Input_Array(I).Longitude;
72.       Dlon_Dist := Dlon * 60.0 * Longitude_Conversion;
73.       Dlon_Dist := Dlon_Dist * Dlon_Dist;
74.
75.       Distance := Ada.Numerics.Elementary_Functions.Sqrt(Dlat_Dist+Dlon_Dist);
76.
77.       Ada.Text_Io.Put("The distance between Aircraft with id ");
78.       Ada.Text_IO.Put(Integer'Image(Input_Array(1).Aircraft_Number));
79.       Ada.Text_Io.Put(" Aircraft with id ");
80.       Ada.Text_Io.Put(Integer'Image(Input_Array(I).Aircraft_Number));
81.       Ada.Text_Io.Put(" is ");
82.       Ada.Text_Io.Put(Float'Image(Distance));
83.       Ada.Text_Io.New_Line;
84.     end loop;
```

```
85.    end Compute_Distances;
86. end My_Aircraft;
87.
```

87 lines: No errors

Main Program

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

Compiling: c:/docume~2/joeb/desktop/16070/codeso~1/test_my_aircraft.adb (source file time stamp: 2003-10-08 18:12:00)

```
 1. ---------------------------------------------------
 2. -- Program to test My_Aircraft
 3. -- Programmer : Joe B
 4. -- Date Last Modified : 10/07/2003
 5. ---------------------------------------------------
 6.
 7. with My_Aircraft;
 8.
 9. procedure Test_My_Aircraft is
10.    Test_Aircraft : My_Aircraft.Aircraft_Array;
11.
12. begin
13.
14.    Test_Aircraft := My_Aircraft.Get_Aircraft_Info;
15.    My_Aircraft.Sort_Aircraft (Test_Aircraft);
16.    My_Aircraft.Compute_Distances(Test_Aircraft);
17.
18. end Test_My_Aircraft;
19.
20.
```

20 lines: No errors