**Actividad Práctica 1**

El objetivo de la actividad es que el alumno pueda compilar, corregir, ejecutar y comprobar si el siguiente código soluciona el problema planteado.

Cuando el alumno **compile** solucionará todos los problemas de sintaxis que tiene el código. Cuando el alumno **corrija** solucionará todos los problemas de lógica que tiene el código. Cabe aclarar que no se deben eliminar ni agregar módulos. Cuando el alumno **ejecute** verificará que los resultados que muestran su solución son los esperados. En caso de no serlo deberá modificar, compilar y ejecutar hasta que los resultados obtenidos sean los esperados

**Enunciado**

Una empresa de energía eléctrica maneja la información de sus clientes, de cada cliente conoce apellido y nombre, dni, categoría y monto que paga del servicio. Se pide:

1. Generar una lista con la información de clientes hasta leer el cliente con apellido y nombre ‘ZZZ’.
2. A partir de la lista generada en a, generar un árbol ordenado por dni con los clientes. Los hijos más chicos deben agregarse a la izquierda y los más grandes o iguales a la derecha.
3. A partir de la estructura generada en b, imprimir los dni de los clientes ordenados de menor a mayor.
4. A partir de la estructura generada en b, imprimir el cliente con mayor categoría.

**Pruebas**

1. Probar el programa si se lee la siguiente información:

ApeyNom: BCD

Dni: 1011

Categoría: 7

ApeyNom: CDA

Dni: 8956

Categoría: 2

ApeyNom: AB

Dni: 4256

Categoría: 5

ApeyNom: AA

Dni: 6523

Categoría: 5

ApeyNom: ZZZ

Dni: 112

Categoría: 25

ApeyNom: GFD

Dni: 691

Categoría: 30

ApeyNom: NDD

Dni: 235

Categoría: 16

ApeyNom ZZ

Dni: 542

Categoría: 3

1. Probar el programa si se lee la siguiente información:

ApeyNom: NDD

Dni: 235

Categoría: 16

ApeyNom ZZ

Dni: 542

Categoría: 3

ApeyNom: BCD

Dni: 1011

Categoría: 7

ApeyNom: CDA

Dni: 8956

Categoría: 56

ApeyNom: AB

Dni: 4256

Categoría: 3

ApeyNom: AA

Dni: 6523

Categoría: 5

ApeyNom: ZZZ

Dni: 112

Categoría: 25

ApeyNom: GFD

Dni: 691

Categoría: 30

1. Probar el programa si no se lee ningún cliente.
2. program actividad1;
3. Type
4. cliente = record
5. apeynom:string;
6. dni:integer;
7. categoria:integer;
8. end;
9. lista = ^nodo;
10. nodo = record
11. dato:cliente;
12. sig:lista;
13. end;
14. arbol= ^nodoA;
15. nodoA= record
16. dato: cliente;
17. HI:arbol:
18. HD:arbol;
19. end;
20. procedure leerCliente (var cli:cliente);
21. begin
22. readln(cli.apeynom);
23. readln(cli.dni);
24. readln(cli.categoria);
25. end;
26. procedure cargarLista (L:lista);
27. var
28. cliente:cliente;
29. aux:nodo;
30. begin
31. leer (cliente);
32. while (apeynom <> 'ZZZ') do
33. begin
34. if (L= nil) then
35. begin
36. new(L);
37. L^.dato:= cliente;
38. L^.sig:= nil;
39. end;
40. else begin
41. new(aux);
42. aux^.sig:= L;
43. L:=aux;
44. end;
45. leerCliente(cliente);
46. end
47. procedure imprimir (var A:arbol);
48. begin
49. if (A <> nil) then
50. begin
51. imprimir (A^.HD);
52. imprimir (A^.HI);
53. writeln (A\*.dato.dni);
54. end;
55. end;
56. procedure agregarEnArbol (A:arbol; c:cliente);
57. begin
58. if (A = nil) then
59. begin
60. new(A);
61. A^.dato:= c;
62. A^.HI:= nil;
63. A^.HD:= nil;
64. end
65. else
66. if (c.dni <= A^.HI.dato) then agregarEnArbol(A^.HI, c);
67. else
68. agregarEnArbol(A^.HD,c);
69. end.
70. procedure generarArbol (var L:lista; var A:arbol);
71. begin
72. while (L^.sig <> nil) then
73. begin
74. agregarEnArbol(A, L.dato);
75. L:= L^.sig;
76. end;
77. end;
78. procedure mayorCategoria (A:arbol; var c:cliente);
79. begin
80. if (A <> nil) then
81. begin
82. if (A\*.HD <> nil) then
83. mayorCategoria (A^.HD,c);
84. else
85. c = A^.dato;
86. end;
87. end;
88. VAR
89. L:lista;
90. A: arbol;
91. c:cliente;
93. BEGIN
94. cargarLista (L);
95. generarArbol(L,A);
96. write ('Muestro los dni ordenados de menor a mayor');
97. mostrarDNI(A);
98. writeln;
99. writeln;
100. mayorCategoria(A,c);
101. write('El cliente',c.apeynom, 'es el de mayor categora');
102. readln;
103. END.