

```

1  #include <iostream>
2  using namespace std;
3  struct nodo
4  {
5      int inf,fe;
6      nodo *izq,*der;
7  };
8  void inserta_AVL(nodo *&raiz,int dato,int cen);
9  void buscar_AVL(nodo *raiz,int dato);
10 void elimina_AVL(nodo *&raiz,int dato);
11 void preorden(nodo *raiz);
12 void inorden(nodo *raiz);
13 void postorden(nodo *raiz);
14 int main(void)
15 {
16     nodo *raiz=NULL;
17     int op,dato;
18     char opc;
19     system("cls");
20     do
21     {
22         system("cls");
23         cout<<"\t\tMENU - ARBOLES BINARIOS BALANCEADOS - AVL\n\n";
24         cout<<"\t\t*****\n\n";
25         cout<<"\t\t1. INSERCIÓN\n\n";
26         cout<<"\t\t2. ELIMINACIÓN\n\n";

```

```

27         cout<<"\t\t3. BUSQUEDA\n\n";
28         cout<<"\t\t4. RECORRIDO PREORDEN\n\n";
29         cout<<"\t\t5. RECORRIDO INORDEN\n\n";
30         cout<<"\t\t6. RECORRIDO POSTORDEN\n\n";
31         cout<<"\t\t7. S A L I R\n\n";
32         cout<<"\t\t\t\t\tOPCIÓN ";
33         cin>>op;
34         system("cls");
35         switch(op)
36         {
37             case 1:
38                 do
39                 {
40                     system("cls");
41                     cout<<"\n\nINGRESE EL DATO A INSERTAR: ";
42                     cin>>dato;
43                     inserta_AVL(raiz,dato,1);
44                     cout<<"\n\n\t\tMAS DATOS? (S/N) : ";
45                     cin>>opc;
46                 }
47                 while(opc=='s' || opc=='S');
48                 system("pause");
49                 break;
50             case 2:
51                 system("cls");
52                 /*cout<<"\n\nINGRESE EL DATO A ELIMINAR: ";

```

```

53     cin>>dato;
54     elimina_abb(raiz,dato);    */
55     system("pause");
56     break;
57 case 3:
58     system("cls");
59     cout<<"\n\nINGRESE EL DATO A BUSCAR: ";
60     cin>>dato;
61     buscar_AVL(raiz,dato);
62     system("pause");
63     break;
64 case 4:
65     system("cls");
66     cout<<"\n\tRECORRIDO PREORDEN";
67     cout<<"\n\t*****\n\n";
68     preorden(raiz);
69     system("pause");
70     break;
71 case 5:
72     system("cls");
73     cout<<"\n\tRECORRIDO INORDEN";
74     cout<<"\n\t*****\n\n";
75     inorden(raiz);
76     system("pause");
77     break;
78 case 6:

```

```

79     system("cls");
80     cout<<"\n\tRECORRIDO POSTORDEN";
81     cout<<"\n\t*****\n\n";
82     postorden(raiz);
83     system("pause");
84     break;
85 }
86 }
87 while(op<6);
88 return(1);
89 }
90 void inserta_AVL(nodo *&raiz,int dato,int cen)
91 {
92     nodo *raiz1,*raiz2,*x;
93     if(raiz!=NULL)
94     {
95         if(dato<raiz->inf)
96         {
97             inserta_AVL(raiz->izq,dato,cen);
98             if(cen==1)
99             {
100                 switch(raiz->fe)
101                 {
102                     case 1:raiz->fe=0;cen=0;break;
103                     case 0:raiz->fe=-1;break;
104                     case -1:

```

```
105
106
107 □
108
109
110
111
112 -
113
114 □
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129 -
130

raiz1=raiz->izq;
if(raiz1->fe<=0)
{
    raiz->izq=raiz1->der;
    raiz1->der=raiz;
    raiz->fe=0;
    raiz=raiz1;
}
else
{
    raiz2=raiz1->der;
    raiz->izq=raiz2->der;
    raiz2->der=raiz;
    raiz1->der=raiz2->izq;
    raiz2->izq=raiz1;
    if(raiz2->fe==-1)
        raiz->fe=1;
    else
        raiz->fe=0;
    if(raiz2->fe==1)
        raiz1->fe=-1;
    else
        raiz1->fe=0;
    raiz=raiz2;
}
raiz->fe=0;
```

```

131         cen=0;
132         break;
133     }
134 }
135 }
136 else if(dato>raiz->inf)
137 {
138     inserta_AVL(raiz->der,dato,cen);
139     if(cen==1)
140     {
141         switch(raiz->fe)
142         {
143             case -1:raiz->fe=0;cen=0;break;
144             case 0:raiz->fe=1;break;
145             case 1:
146                 raiz1=raiz->der;
147                 if(raiz1->fe>=0)
148                 {
149                     raiz->der=raiz1->izq;
150                     raiz1->izq=raiz;
151                     raiz->fe=0;
152                     raiz=raiz1;
153                 }
154                 else
155                 {
156                     raiz2=raiz1->izq;
157                     raiz->der=raiz2->izq;
158                     raiz2->izq=raiz;
159                     raiz1->izq=raiz2->der;
160                     raiz2->der=raiz1;
161                     if(raiz2->fe==1)
162                         raiz->fe=-1;
163                     else
164                         raiz->fe=0;
165                     if(raiz2->fe==-1)
166                         raiz1->fe=1;
167                     else
168                         raiz1->fe=0;
169                     raiz=raiz2;
170                 }
171                 raiz->fe=0;
172                 cen=0;
173                 break;
174             }
175         }
176     }
177 else
178     cout<<"\nEL DATO YA EXISTE ... \n\n";
179 }
180 else
181 {
182     x=new(nodo);

```

```

183         x->inf=dato;
184         x->izq=NULL;
185         x->der=NULL;
186         x->fe=0;
187         raiz=x;
188         cen=1;
189     }
190 }
191 void buscar_AVL(nodo *raiz,int dato)
192 {
193     if(dato<raiz->inf)
194         if(raiz->izq==NULL)
195             cout<<"\n\tEL DATO BUSCADO NO EXISTE..... \n\n";
196         else
197             buscar_AVL(raiz->izq,dato);
198     else
199         if(dato>raiz->inf)
200             if(raiz->der==NULL)
201                 cout<<"\n\tEL DATO BUSCADO NO EXISTE.....\n\n ";
202             else
203                 buscar_AVL(raiz->der,dato);
204         else
205             cout<<"\n\tEL DATO BUSCADO SI EXISTE.....\n\n";
206 }
207 void elimina_AVL(nodo *&raiz,int dato)
208 {

```

```

209
210
211 }
212 void preorden(nodo *raiz)
213 {
214     if(raiz!=NULL)
215     {
216         cout<<raiz->inf<<"\n\n";
217         preorden(raiz->izq);
218         preorden(raiz->der);
219     }
220 }
221 void inorden(nodo *raiz)
222 {
223     if(raiz!=NULL)
224     {
225         inorden(raiz->izq);
226         cout<<raiz->inf<<"\n\n";
227         inorden(raiz->der);
228     }
229 }
230 void postorden(nodo *raiz)
231 {
232     if(raiz!=NULL)
233     {
234         postorden(raiz->izq);
235
236         postorden(raiz->der);
237         cout<<raiz->inf<<"\n\n";
238     }

```