

```

1  #include <iostream>
2  using namespace std;
3  struct nodo
4  {
5      int inf;
6      nodo *izq,*der;
7  };
8  void inserta_abb(nodo *&raiz,int dato);
9  void buscar_abb(nodo *raiz,int dato);
10 void elimina_abb(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\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_abb(raiz,dato);
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_abb(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_abb(nodo *&raiz,int dato)
91 {
92     nodo *q;
93     if(raiz!=NULL)
94         if(dato<raiz->inf)
95             inserta_abb(raiz->izq,dato);
96         else
97             if(dato>raiz->inf)
98                 inserta_abb(raiz->der,dato);
99             else
100                 cout<<"\n\tEL DATO YA SE ENCUENTRA EN EL ARBOL....\n ";
101     else
102     {
103         q=new(nodo);
104         q->izq=NULL;

```

```

105     q->der=NULL;
106     q->inf=dato;
107     raiz=q;
108 }
109 }
110 void buscar_abb(nodo *raiz,int dato)
111 {
112     if(dato<raiz->inf)
113         if(raiz->izq==NULL)
114             cout<<"\n\tEL DATO BUSCADO NO EXISTE..... \n\n";
115         else
116             buscar_abb(raiz->izq,dato);
117     else
118         if(dato>raiz->inf)
119             if(raiz->der==NULL)
120                 cout<<"\n\tEL DATO BUSCADO NO EXISTE.....\n\n ";
121             else
122                 buscar_abb(raiz->der,dato);
123         else
124             cout<<"\n\tEL DATO BUSCADO SI EXISTE.....\n\n";
125 }
126 void elimina_abb(nodo *&raiz,int dato)
127 {
128     nodo *q,*aux,*aux1;
129     int cen;
130     if(raiz!=NULL)
131     {
132         if(dato<raiz->inf)
133             elimina_abb(raiz->izq,dato);
134         else
135             if(dato>raiz->inf)
136                 elimina_abb(raiz->der,dato);
137         else
138         {
139             q=raiz;
140             if(q->der==NULL)
141                 raiz=q->izq;
142             else
143                 if(q->izq==NULL)
144                     raiz=q->der;
145                 else
146                 {
147                     aux=q->izq;
148                     cen=0;
149                     while(aux->der!=NULL)
150                     {
151                         aux1=aux;
152                         aux=aux->der;
153                         cen=1;
154                     };
155                     raiz->inf=aux->inf;
156                     q=aux;

```

```

157         if(cen==1)
158             aux1->der=aux->izq;
159         else
160             raiz->izq=aux->izq;
161     }
162     delete(q);
163 }
164 }
165 }
166 void preorden(nodo *raiz)
167 {
168     if(raiz!=NULL)
169     {
170         cout<<raiz->inf<<"\n\n";
171         preorden(raiz->izq);
172         preorden(raiz->der);
173     }
174 }
175 void inorden(nodo *raiz)
176 {
177     if(raiz!=NULL)
178     {
179         inorden(raiz->izq);
180         cout<<raiz->inf<<"\n\n";
181         inorden(raiz->der);
182     }
183 }
184 void postorden(nodo *raiz)
185 {
186     if(raiz!=NULL)
187     {
188         postorden(raiz->izq);
189         postorden(raiz->der);
190         cout<<raiz->inf<<"\n\n";
191     }
192 }

```