```
1 #include "bintree.h"
 2 #include <iostream>
 3
 4 using namespace std;
 5
 6 template <class T>
 7 void postorden(const bintree<T> &A, const typename bintree<T>::node &v)
9
       if (!v.null())
10
       {
           postorden(A, v.left());
11
12
           postorden(A, v.right());
13
           cout << *v;
14
       }
15 }
16
17 string postorden_pre_ino(string pre,string ino)
18 {
19
       string post="";
20
       string aux1,aux2;
21
22
       int raiz=ino.find_first_of(pre[0]);
23
24
       if(raiz!=0) //SI HAY ARBOL IZQUIERDO
25
       {
26
           aux1=pre.substr(1,raiz);
27
           aux2=ino.substr(0,raiz);
           //cout << "ARBOL IZQUIERDO ->>>> " << aux1 << "<->" << aux2 << endl;
28
29
           post+=postorden_pre_ino(aux1,aux2);
30
       }
       if(raiz!=ino.length()-1) //SI HAY ARBOL DERECHO
31
32
33
           aux1=pre.substr(raiz+1,pre.length()-1);
           aux2=ino.substr(raiz+1,ino.length()-1);
34
           //cout << "ARBOL DERECHO ->>>> " << aux1 << "<->" << aux2 << endl;
35
36
           post+=postorden pre ino(aux1,aux2);
37
       }
38
39
       post+=pre[0];
40
41
       return post;
42
43 }
44
45
46 int main()
47 {
48
       bintree<char> arb('G');
49
50
       //RAMA IZQUIERDA
51
       arb.insert_left(arb.root(), 'E');
52
53
       bintree<char>::node node_aux = arb.root().left();
54
55
       arb.insert_left(node_aux, 'A');
56
57
       node_aux = node_aux.left();
58
59
       arb.insert left(node aux, 'I');
       arb.insert_right(node_aux, 'B');
60
```

14/1/2020 ejercicio04.cpp 61 62 //RAMA DERECHA 63 arb.insert\_right(arb.root(), 'M'); 64 65 node\_aux = arb.root().right(); 66 arb.insert\_left(node\_aux, 'C'); 67 arb.insert\_right(node\_aux, 'K'); 68 69 node\_aux = node\_aux.left(); 70 arb.insert\_left(node\_aux, 'L'); 71 arb.insert\_right(node\_aux, 'F'); 72 73 74 node\_aux = node\_aux.left(); arb.insert\_right(node\_aux, 'D'); 75 76 node\_aux = arb.root().right().right(); 77 78 arb.insert\_right(node\_aux, 'J'); 79 node\_aux = node\_aux.right(); 80 arb.insert\_left(node\_aux, 'H'); 81 82 postorden(arb,arb.root()); 83 84 cout << endl;</pre> 85 string preorder=postorden\_pre\_ino("GEAIBMCLDFKJH","IABEGLDCFMKHJ");

cout << preorder << endl;</pre>

86 87 }

localhost:4649/?mode=clike 2/2