

```
1 #include <iostream>
2 #include <string>
3 #include <list>
4
5 using namespace std;
6
7 template <typename T>
8 void mostrar_lista(const list<T> &l)
9 {
10     typename list<T>::const_iterator p;
11     for (p = l.begin(); p != l.end(); p++)
12     {
13         cout << *p << " ";
14     }
15     cout << endl;
16 }
17
18 template <typename T>
19 list<T> mezclar(const list<T> &l1, const list<T> &l2)
20 {
21     list<T> ret;
22     T aux;
23     typename list<T>::const_iterator p;
24     typename list<T>::const_iterator q;
25     p = l1.begin();
26     q = l2.begin();
27     while (p != l1.end() && q != l2.end())
28     {
29         if (*p < *q)
30         {
31             aux = *p;
32             p++;
33         }
34         else
35         {
36             aux = *q;
37             q++;
38         }
39
40         ret.push_back(aux);
41     }
42     if (p == l1.end())
43     {
44         ret.push_back(*q);
45         q++;
46     }
47     else
48     {
49         ret.push_back(*p);
50         p++;
51     }
52     return ret;
53 }
54 int main()
55 {
56     list<int> aux;
57     aux.push_back(0);
58     aux.push_back(1);
59     aux.push_back(1);
60     aux.push_back(1);
```

```
61     aux.push_back(1);
62     aux.push_back(3);
63     aux.push_back(8);
64     list<int> aux2;
65     aux2.push_back(-1);
66     aux2.push_back(2);
67     aux2.push_back(7);
68     aux2.push_back(10);
69     aux2.push_back(1);
70     aux2.push_back(3);
71     aux2.push_back(8);
72     mostrar_lista(aux);
73     mostrar_lista(aux2);
74     list<int> aux3;
75     aux3 = mezclar(aux, aux2);
76     mostrar_lista(aux3);
77 }
```