```
1 #include <iostream>
 2 #include <string>
 3 #include <list>
 5 using namespace std;
 6
 7 template <typename T>
 8 void invertir orden(list<T> &l)
9 {
10
       typename list<T>::iterator final=l.end();
11
12
       final--;
13
       typename list<T>::iterator principio=l.begin();
       while (final!=principio)
14
15
16
           aux=*principio;
17
           *principio=*final;
           *final=aux;
18
19
           principio++;
           if(principio!=final)
20
21
22
               final--;
23
           }
24
       }
25
26 }
27
28 template <typename T>
29 void mostrar_lista(const list<T> &1)
30 {
       typename list<T>::const_iterator p;
31
32
       for (p = 1.begin(); p != 1.end(); p++)
33
       {
34
           cout << *p << " ";
35
       cout << endl;</pre>
36
37 }
38
39 int main()
40 {
41
       list<int> aux;
42
       aux.push_back(2);
43
       aux.push back(10);
       aux.push_back(14);
44
45
       aux.push_back(13);
46
       aux.push back(1);
47
       aux.push back(5);
48
       aux.push_back(3);
49
       aux.push_back(80);
50
       mostrar_lista(aux);
       invertir orden(aux);
51
       mostrar lista(aux);
52
53 }
```

localhost:4649/?mode=clike 1/1