```cpp
#include <iostream>
#include <string>
#include <list>

using namespace std;

template <typename T> using Lista_Comprimida=list<pair<T,int>>;

template <typename T>
void mostrar_lista(const list<T> &l)
{
    typename list<T>::const_iterator p;
    for (p = l.begin(); p != l.end(); p++)
    {
        cout << *p << " ";
    }
    cout << endl;
}


template <typename T>
void mostrar_lista(const Lista_Comprimida<T> &l)
{
    typename Lista_Comprimida<T>::const_iterator p;
    for (p = l.begin(); p != l.end(); p++)
    {
        cout << (*p).first << " " << (*p).second << endl;
    }
}


template <typename T>
Lista_Comprimida<T> comprimir (const list<T> &l)
{

    int contador=0;
    Lista_Comprimida<T> ret;
    T aux;

    typename list<T>::const_iterator it;
    it=l.cbegin();
    aux=*it;
    for(it;it!=l.cend();it++)
    {
        if(aux==(*it))
        {
            contador++;
        }else
        {
            ret.push_back(make_pair(aux,contador));
            aux=(*it);
            contador=0;
        }
    }
    ret.push_back(make_pair(aux,contador));

    return ret;

}

```

```
61 template <typename T>
62 list <T> descomprimir (const Lista_Comprimida<T> &lc)
63 {
64     typename Lista_Comprimida<T>::const_iterator it;
65     it=lc.cbegin();
66     list<T> ret((*it).second,(*it).first);
67     typename list<T>::iterator p;
68     it++;
69     for(it;it!=lc.cend();it++)
70     {
71         p=ret.end();
72         ret.insert(p,(*it).second,(*it).first);
73     }
74     return ret;
75 
76 }
77 
78 int main()
79 {
80     cout << "Prueba 1 " << endl;
81     Lista_Comprimida<int> lc1;
82     pair<int,int> comp1,comp2,comp3;
83     comp1.first=2;
84     comp1.second=4;
85     comp2.first=3;
86     comp2.second=5;
87     comp3.first=7;
88     comp3.second=2;
89     lc1.push_back(comp1);
90     lc1.push_back(comp2);
91     lc1.push_back(comp3);
92     list<int> ld1;
93     ld1=descomprimir(lc1);
94     mostrar_lista(lc1);
95     mostrar_lista(ld1);
96 
97 
98     cout << "Prueba 2" << endl;
99     list<int> l2;
100    l2.push_back(8);
101    l2.push_back(8);
102    l2.push_back(8);
103    l2.push_back(8);
104    l2.push_back(8);
105 
106    l2.push_back(2);
107    l2.push_back(2);
108    l2.push_back(2);
109 
110    l2.push_back(4);
111    l2.push_back(4);
112    l2.push_back(4);
113    l2.push_back(4);
114    Lista_Comprimida<int> lc2;
115    lc2=comprimir(l2);
116    mostrar_lista(l2);
117    mostrar_lista(lc2);
118 
119 }
```