

```
1 #include <iostream>
2 #include <string>
3 #include <list>
4 #include <vector>
5
6 using namespace std;
7
8 template <typename T> using Par_INT_T = pair<int,T>;
9
10 template <typename T>
11 class vdisperso
12 {
13 private:
14     list<pair<int, T>> coefs;
15     int n;
16
17 public:
18     vdisperso(const vector<T> &v);
19     void asignar_coeficiente(int i, const T &x);
20     vector<T> convertir() const;
21     void mostrarVectorDis() const;
22 };
23
24 template <typename T>
25 void vdisperso<T>::mostrarVectorDis() const{
26     for(typename list<pair<int,T>>::const_iterator
27 it=coefs.cbegin();it!=coefs.cend();it++)
28     {
29         cout << "POSICION:" << (*it).first << "->" << (*it).second << endl;
30     }
31 }
32
33 template <typename T>
34 vdisperso<T>::vdisperso(const vector<T> &v)
35 {
36     int i;
37     pair<int,T> aux;
38     this->n=v.size();
39     for(i=0;i<v.size();i++)
40     {
41         if(v[i]!=T())
42         {
43             aux.first=i;
44             aux.second=v[i];
45             this->coefs.push_back(aux);
46         }
47     }
48 }
49
50 template <typename T>
51 void vdisperso<T>::asignar_coeficiente(int i, const T &x)
52 {
53     typename list<pair<int, T>>::iterator it;
54     it=this->coefs.begin();
55     while(it!=this->coefs.end() && (*it).first!=i)
56     {
57         it++;
58     }
59     if(it!=this->coefs.end())
```

```
60     {
61         this->coefs.erase(it);
62     }
63     pair<int, T> aux;
64     aux.first=i;
65     aux.second=x;
66     this->coefs.push_back(aux);
67
68 }
69
70 template <typename T>
71 vector<T> vdisperso<T>::convertir() const
72 {
73     vector<T> ret;
74     typename list<pair<int,T>>::const_iterator it;
75     it=this->coefs.begin();
76     for(int i=0;i<this->n;i++)
77     {
78         if(i==(*it).first)
79         {
80             ret.push_back((*it).second);
81             it++;
82         }else
83         {
84             ret.push_back(T());
85         }
86     }
87     return ret;
88 }
89
90 template <typename T>
91 void mostrar_vector(const vector<T> & v)
92 {
93     for(typename vector<T>::const_iterator it=v.cbegin();it!=v.cend();it++)
94     {
95         cout << (*it) << " ";
96     }
97     cout << endl;
98 }
99
100 int main()
101 {
102     vector<int> aux(10,int());
103     aux[0]=104353;
104     aux[2]=3;
105     aux[4]=9;
106     aux[7]=99;
107     aux[9]=81;
108     vdisperso<int> prueba(aux);
109     prueba.mostrarVectorDis();
110     aux=prueba.convertir();
111     mostrar_vector(aux);
112 }
```