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

using namespace std;
template <typename T> using Lista_iteradores = list<typename
list<T>::const_iterator>;

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>
list<T> obtener_elem(const list<T> &l, const Lista_iteradores<T> &l1)
{
    typename Lista_iteradores<T>::const_iterator p;
    list<T> ret;
    for(p=l1.cbegin();p!=l1.cend();p++)
    {
        ret.push_back(**p);
    }
    return ret;
}

int main()
{
    list<int> prueba1;
    prueba1.push_back(1);
    prueba1.push_back(3);
    prueba1.push_back(4);
    prueba1.push_back(5);
    prueba1.push_back(8);
    Lista_iteradores<int> aux;
    list<int>::const_iterator p=prueba1.cbegin();
    aux.push_back(p); //GUARDAMOS ITERADOR HACIA EL PRIMER ELEMENTO
    p++;
    p++;
    aux.push_back(p); //GUARDAMOS ITERADOR HACIA EL TERCERO
    aux.push_back(p); //GUARDAMOS DOS VECES EL MISMO
    p++;
    p++;
    aux.push_back(p); //GUARDAMOS ITERADOR EN ESTE CASO HACIA EL ULTIMO
    list<int> resultado;
    resultado=obtener_elem(prueba1,aux);
    mostrar_lista(resultado);
}
```