

Relación STL

1. Definir una función que permita invertir un objeto de tipo list. Los elementos que contiene la lista son enteros.

```
void Invertir(const list<int> &lsource, list<int> &ldestino){
    list<int>::const_iterator it = --lsource.end();
    for (; it != lsource.begin(); --it) {
        ldestino.push_back(*it);
    }
    ldestino.push_back(lsource.front());
}
```

2. Definir una función que obtenga una cola de prioridad (priority_queue) con la información de todos los alumnos, de manera que la prioridad se define de mayor a menor valor de selectividad. Así la función sería:

```
#include <iostream>
#include <list>
#include <queue>
#include <vector>

using namespace std;

struct alumno{
    string dni;
    string nombre;
    string apellidos;
    string correo;
    double nota_selectividad;
};

struct Comparator
{
    bool operator()(const alumno &source_alumno, const alumno &source2_alumno){
        return source_alumno.nota_selectividad > source2_alumno.nota_selectividad;
    }
};

void ObtenerPrioridad (const list<alumno> &
alumnos, priority_queue<alumno, vector<alumno>, Comparator> & pq){
    list<alumno>::const_iterator it = alumnos.begin();
    for (; it != alumnos.end(); ++it) {
        pq.push(*it);
    }
}
```

3. Dada la clase list instanciada a enteros, crear una función que elimine los elementos pares de la lista. Para implementarla hacer uso de iteradores.

```
void BorrarPares(list<int> &lsource){
    list<int>::iterator it = lsource.begin();
    while (it != lsource.end()) {
        if ( (*it)%2 == 0)
            it = lsource.erase(it);
        else
            ++it;
    }
}
```

4.