```cpp
#include <iostream>
#include <fstream>
#include <string>
#include "list_eda.h"
#include "persona.h"

using namespace std;

class filtro_edad {

public:

    const int edadMax, edadMin;

    filtro_edad(int min, int max) : edadMin(min), edadMax(max) {};

    bool operator() (const persona& persona) {

        if (persona.getEdad() < edadMin || persona.getEdad() > edadMax) {
            return true;
        }
        else {
            return false;
        }
    }
};

template <class T, class Predicate>
void remove_if(list <T >& lista, Predicate pred) {

    auto it = lista.begin();

    while (it != lista.end()) {

        if (pred(*it)) {

            it = lista.erase(it);
        }
        else {
            ++it;
        }
    }
}



bool resuelveCaso() {

    int n, min, max, edad;
    string nombre;

    cin >> n >> min >> max;

    if (n == 0) return false;
```

```
    list<persona> lista;

    for (int i = 0; i < n; i++) {

        cin >> edad;
        getline(cin, nombre);
        persona persona(edad, nombre);
        lista.push_back(persona);
    }

    remove_if(lista, filtro_edad(min, max));

    for (auto it = lista.begin(); it != lista.end(); ++it) {
        cout << it->getNombre() << "\n";
    }

    cout << "---" << endl;

    return true;
}


int main() {
# ifndef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf()); // save old buf and redirect std :: cin to
casos .txt
# endif


    while (resuelveCaso()) {}

# ifndef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system(" PAUSE ");
# endif
    return 0;
}
```