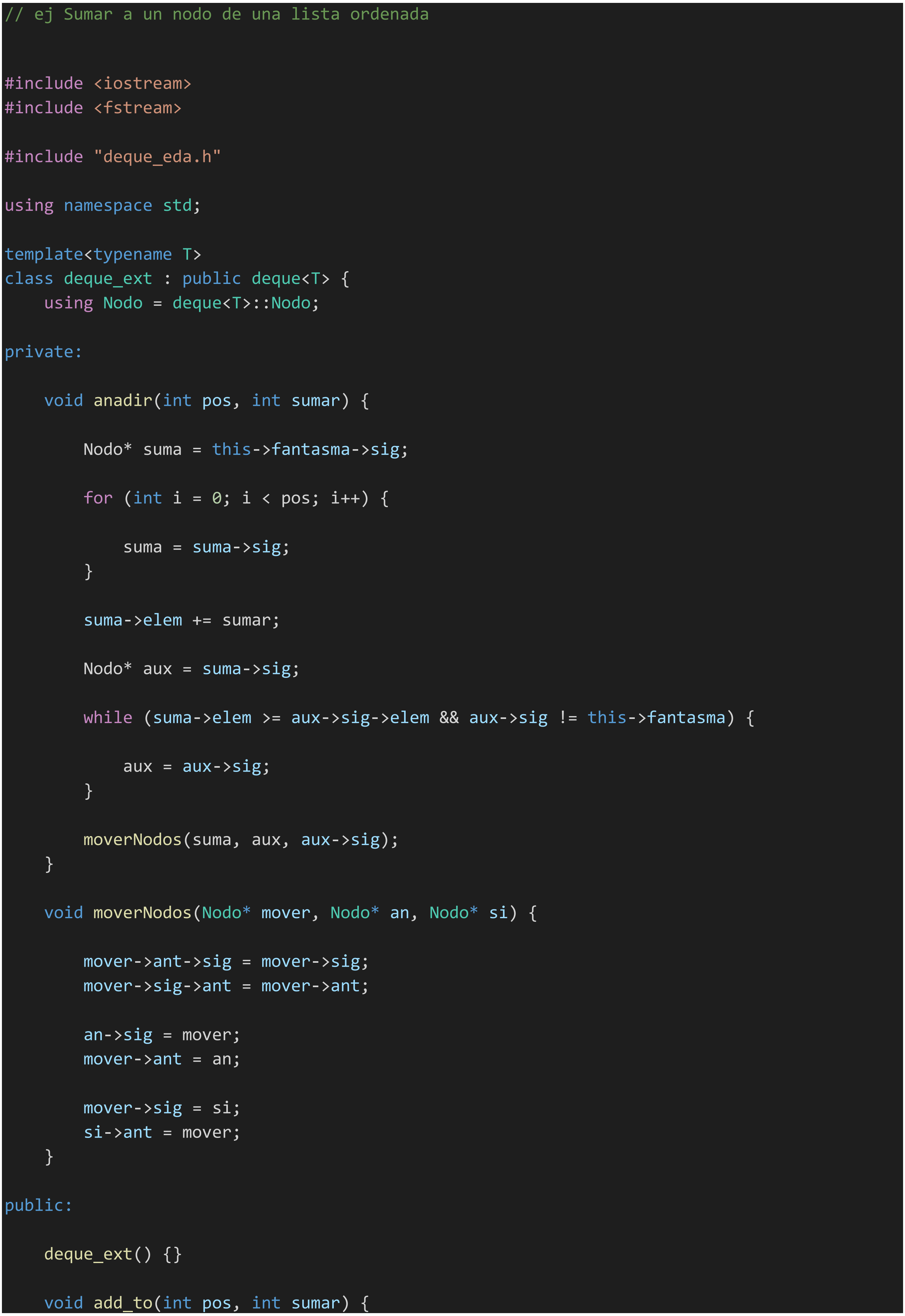
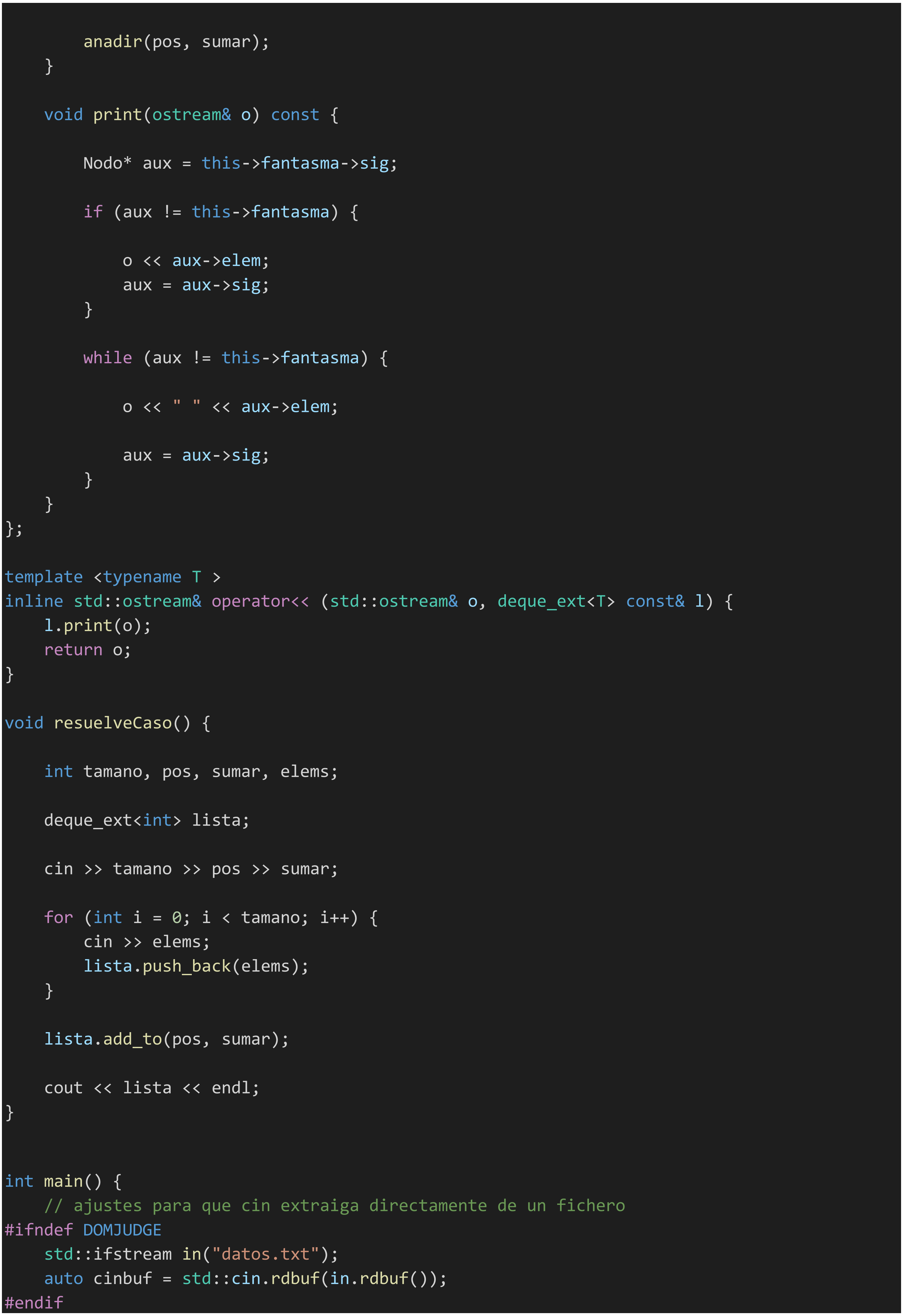


```
// ej Sumar a un nodo de una lista ordenada


#include <iostream>
#include <fstream>

#include "deque_eda.h"

using namespace std;

template<typename T>
class deque_ext : public deque<T> {
    using Nodo = deque<T>::Nodo;

private:

    void anadir(int pos, int sumar) {

        Nodo* suma = this->fantasma->sig;

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

            suma = suma->sig;
        }

        suma->elem += sumar;

        Nodo* aux = suma->sig;

        while (suma->elem >= aux->sig->elem && aux->sig != this->fantasma) {

            aux = aux->sig;
        }

        moverNodos(suma, aux, aux->sig);
    }

    void moverNodos(Nodo* mover, Nodo* an, Nodo* si) {

        mover->ant->sig = mover->sig;
        mover->sig->ant = mover->ant;

        an->sig = mover;
        mover->ant = an;

        mover->sig = si;
        si->ant = mover;
    }

public:

    deque_ext() {}

    void add_to(int pos, int sumar) {
```



```cpp
        anadir(pos, sumar);
    }

    void print(ostream& o) const {

        Nodo* aux = this->fantasma->sig;

        if (aux != this->fantasma) {

            o << aux->elem;
            aux = aux->sig;
        }

        while (aux != this->fantasma) {

            o << " " << aux->elem;

            aux = aux->sig;
        }
    }
};

template <typename T >
inline std::ostream& operator<< (std::ostream& o, deque_ext<T> const& l) {
    l.print(o);
    return o;
}

void resuelveCaso() {

    int tamano, pos, sumar, elems;

    deque_ext<int> lista;

    cin >> tamano >> pos >> sumar;

    for (int i = 0; i < tamano; i++) {
        cin >> elems;
        lista.push_back(elems);
    }

    lista.add_to(pos, sumar);

    cout << lista << endl;
}


int main() {
    // ajustes para que cin extraiga directamente de un fichero
#ifndef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif
```

```
    int casos = 0;

    cin >> casos;

    for (int i = 0; i < casos; i++) {
        resuelveCaso();
    }

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