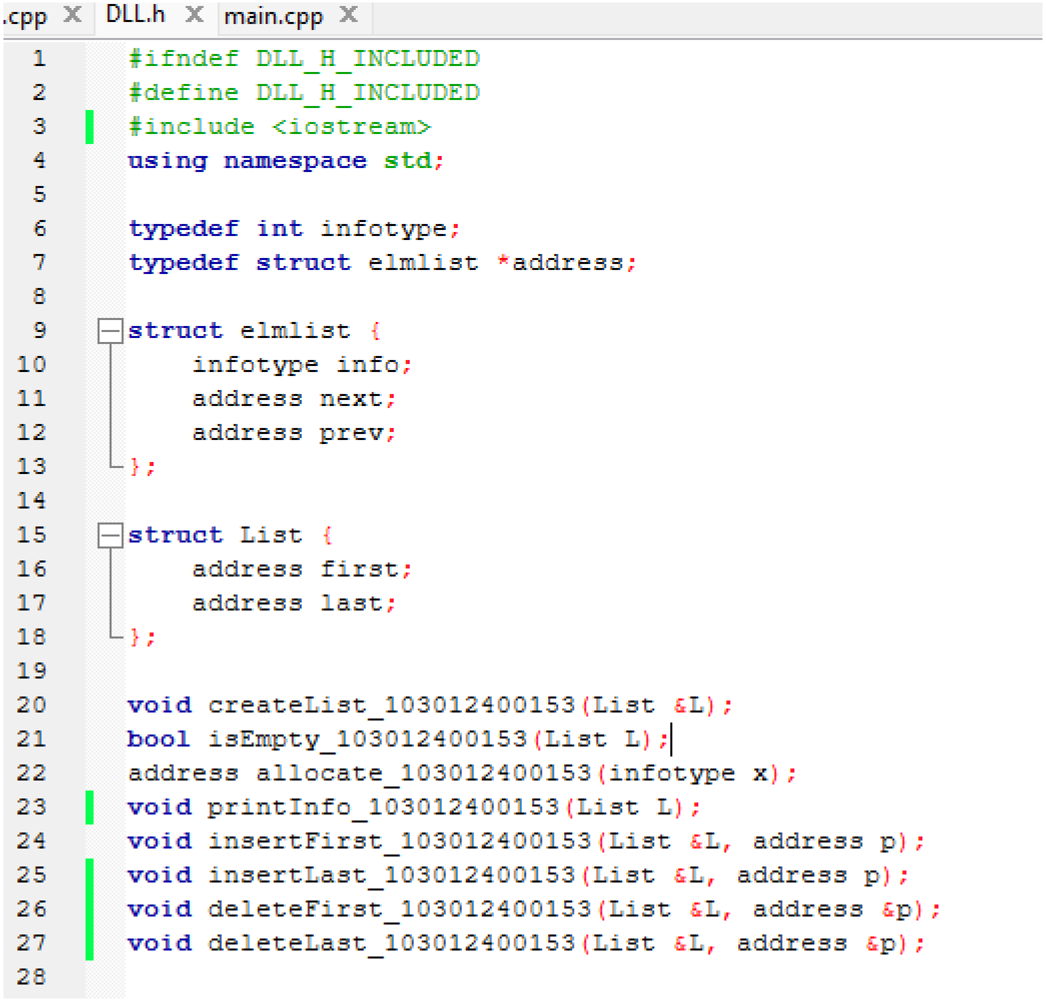
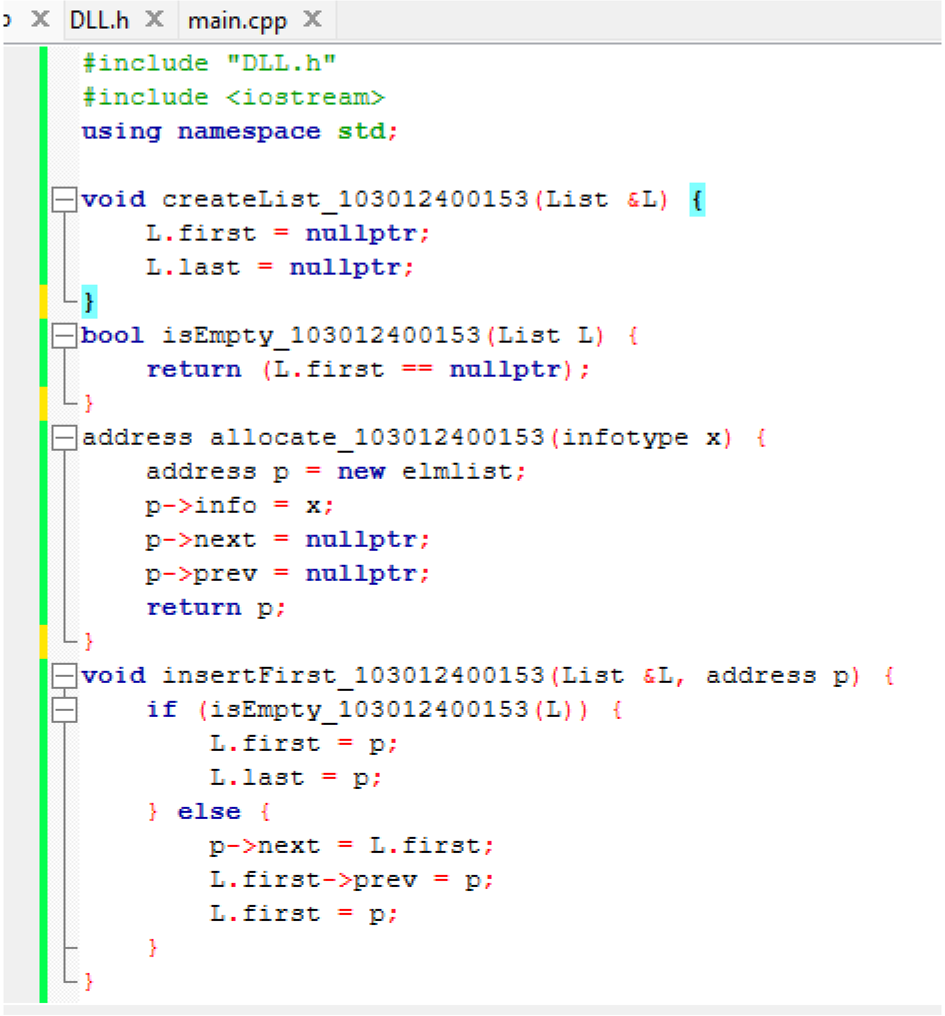
AHMAD RAFIANSYAH

103012400153

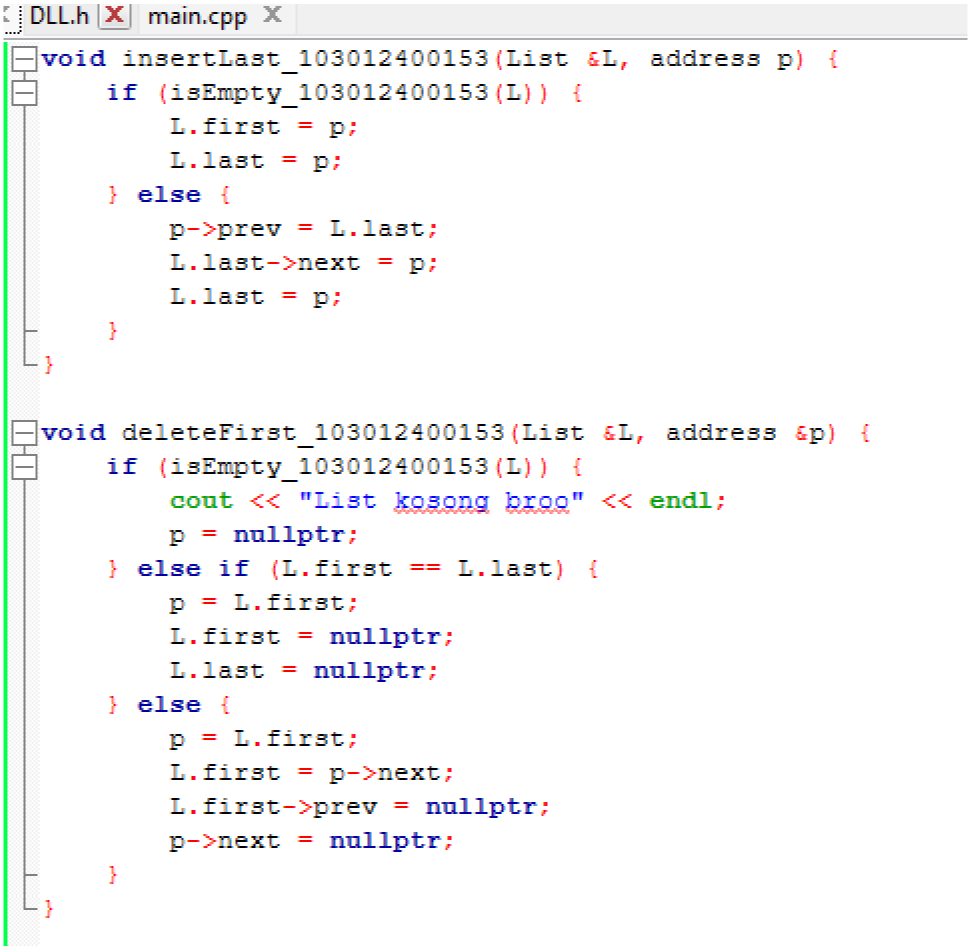
IF-48-11



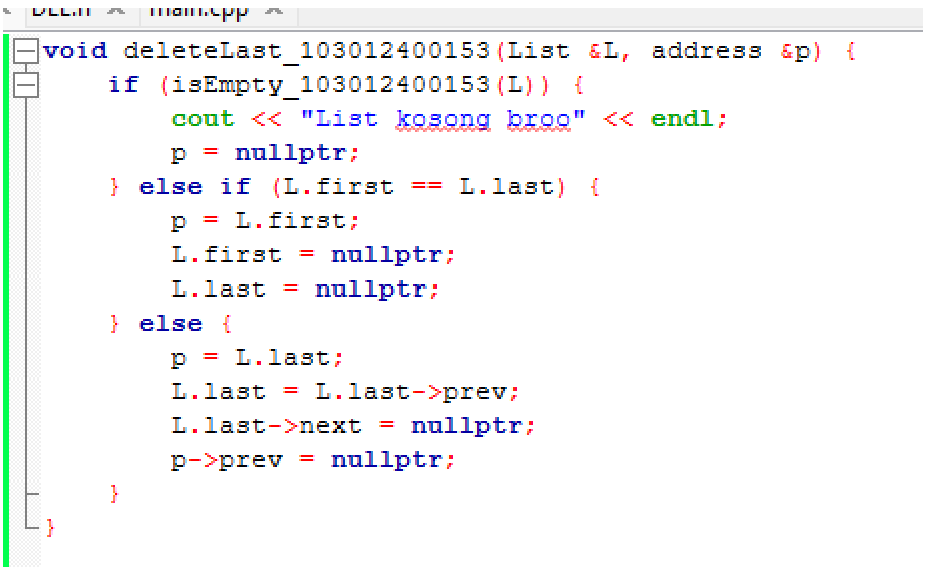
```
#ifndef DLL_H_INCLUDED
#define DLL_H_INCLUDED
#include <iostream>
using namespace std;

typedef int infotype;
typedef struct elmlist *address;

struct elmlist {
    infotype info;
    address next;
    address prev;
};

struct List {
    address first;
    address last;
};

void createList_103012400153(List &L);
bool isEmpty_103012400153(List L);
address allocate_103012400153(infotype x);
void printInfo_103012400153(List L);
void insertFirst_103012400153(List &L, address p);
void insertLast_103012400153(List &L, address p);
void deleteFirst_103012400153(List &L, address &p);
void deleteLast_103012400153(List &L, address &p);

```



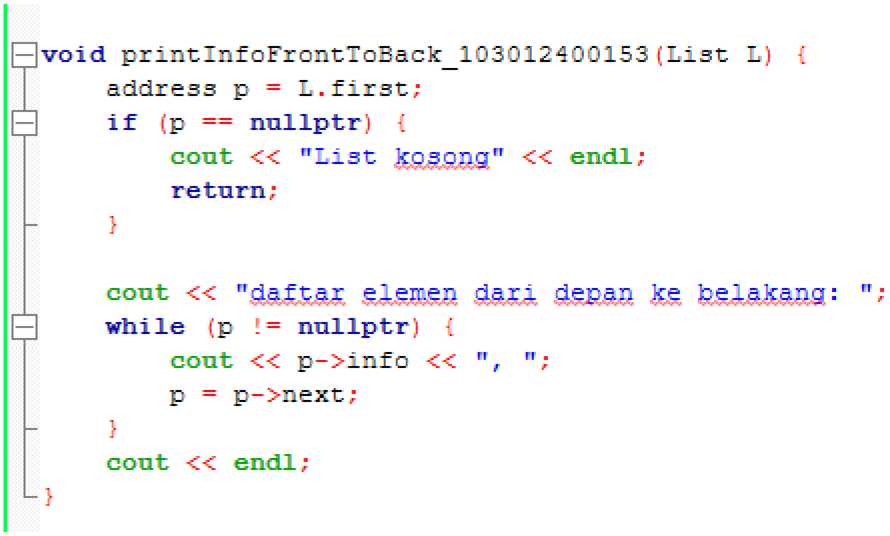
```cpp
#include "DLL.h"
#include <iostream>
using namespace std;

void createList_103012400153(List &L) {
    L.first = nullptr;
    L.last = nullptr;
}
bool isEmpty_103012400153(List L) {
    return (L.first == nullptr);
}
address allocate_103012400153(infotype x) {
    address p = new elmlist;
    p->info = x;
    p->next = nullptr;
    p->prev = nullptr;
    return p;
}
void insertFirst_103012400153(List &L, address p) {
    if (isEmpty_103012400153(L)) {
        L.first = p;
        L.last = p;
    } else {
        p->next = L.first;
        L.first->prev = p;
        L.first = p;
    }
}
```



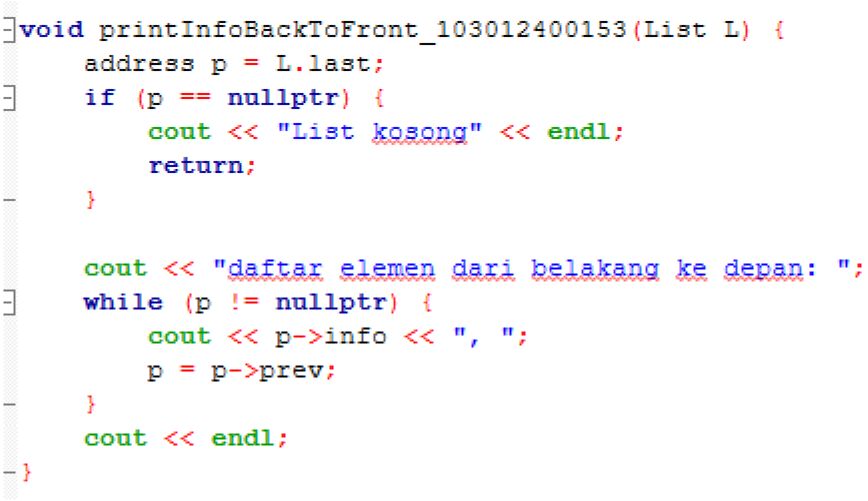
```cpp
void insertLast_103012400153(List &L, address p) {
    if (isEmpty_103012400153(L)) {
        L.first = p;
        L.last = p;
    } else {
        p->prev = L.last;
        L.last->next = p;
        L.last = p;
    }
}

void deleteFirst_103012400153(List &L, address &p) {
    if (isEmpty_103012400153(L)) {
        cout << "List kosong broo" << endl;
        p = nullptr;
    } else if (L.first == L.last) {
        p = L.first;
        L.first = nullptr;
        L.last = nullptr;
    } else {
        p = L.first;
        L.first = p->next;
        L.first->prev = nullptr;
        p->next = nullptr;
    }
}
```



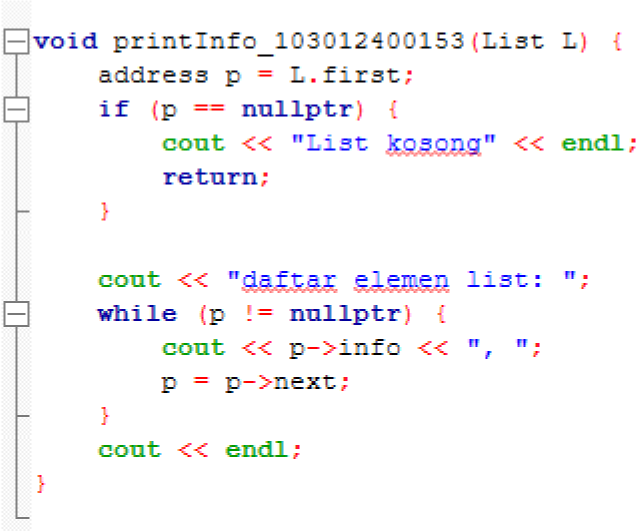
```cpp
void deleteLast_103012400153(List &L, address &p) {
    if (isEmpty_103012400153(L)) {
        cout << "List kosong broo" << endl;
        p = nullptr;
    } else if (L.first == L.last) {
        p = L.first;
        L.first = nullptr;
        L.last = nullptr;
    } else {
        p = L.last;
        L.last = L.last->prev;
        L.last->next = nullptr;
        p->prev = nullptr;
    }
}
```



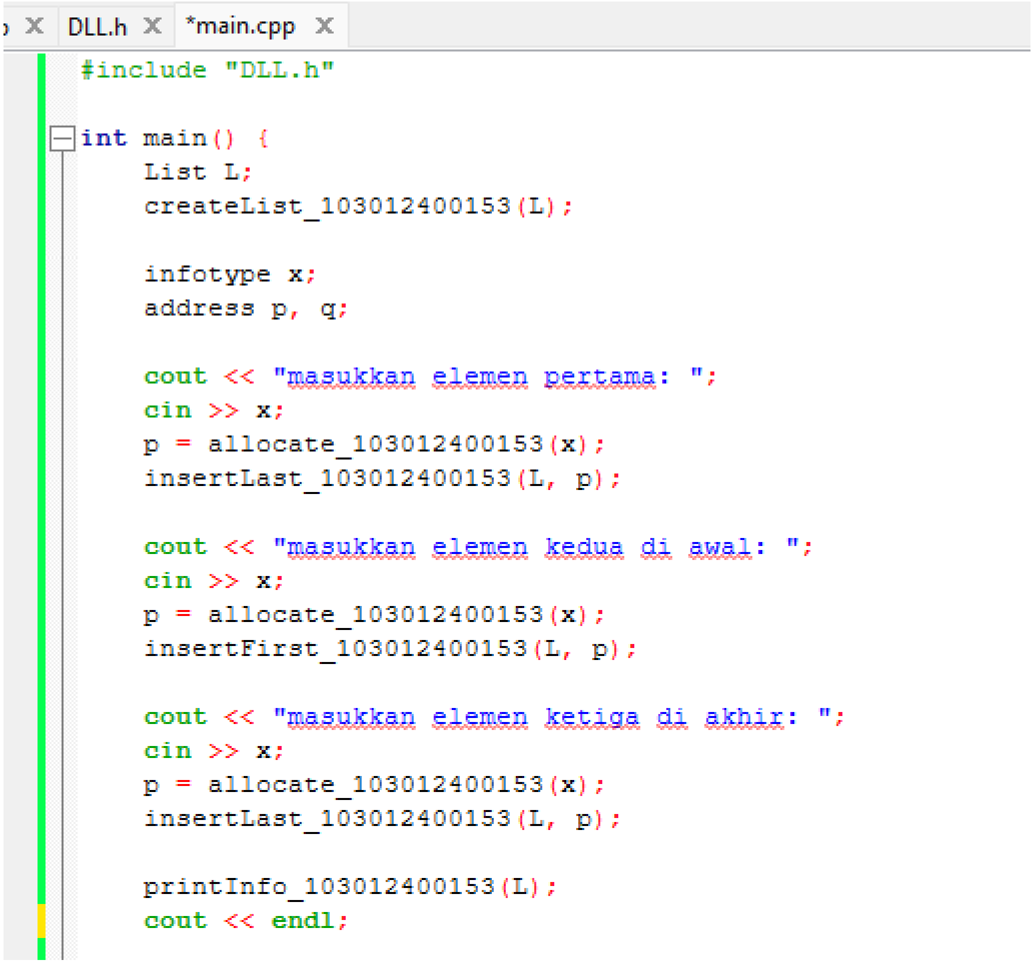
```cpp
void printInfoFrontToBack_103012400153(List L) {
    address p = L.first;
    if (p == nullptr) {
        cout << "List kosong" << endl;
        return;
    }

    cout << "daftar elemen dari depan ke belakang: ";
    while (p != nullptr) {
        cout << p->info << ", ";
        p = p->next;
    }
    cout << endl;
}
```



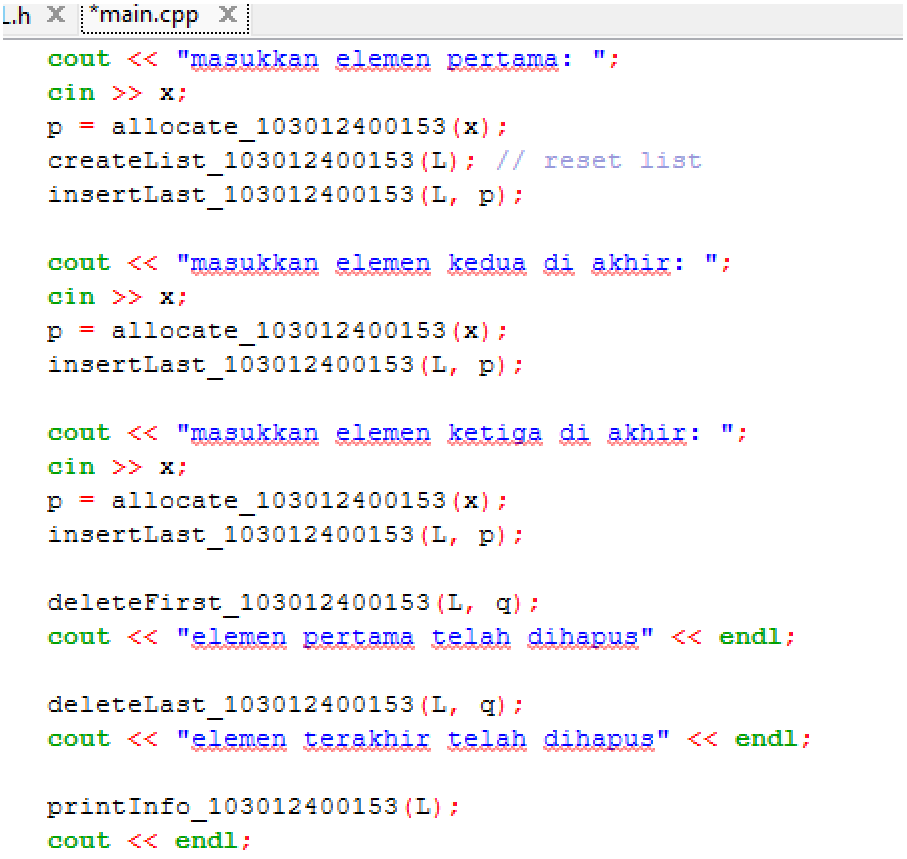
```cpp
void printInfoBackToFront_103012400153(List L) {
    address p = L.last;
    if (p == nullptr) {
        cout << "List kosong" << endl;
        return;
    }

    cout << "daftar elemen dari belakang ke depan: ";
    while (p != nullptr) {
        cout << p->info << ", ";
        p = p->prev;
    }
    cout << endl;
}
```



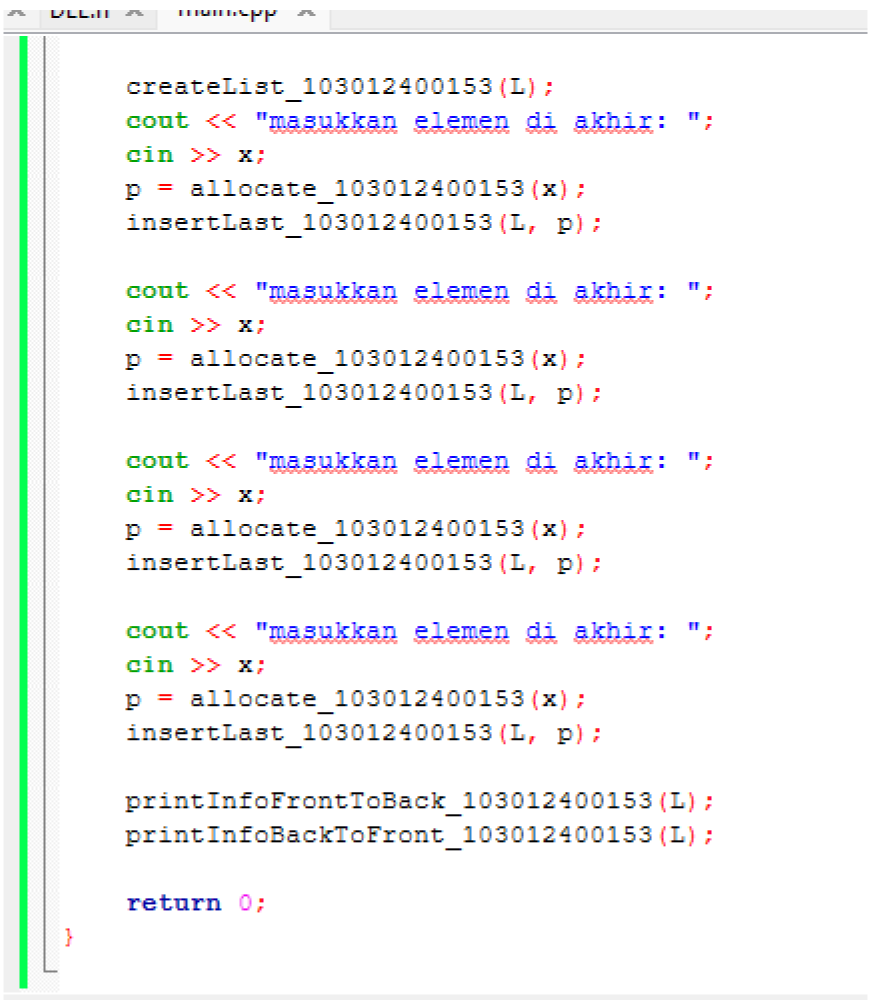
```cpp
void printInfo_103012400153(List L) {
    address p = L.first;
    if (p == nullptr) {
        cout << "List kosong" << endl;
        return;
    }

    cout << "daftar elemen list: ";
    while (p != nullptr) {
        cout << p->info << ", ";
        p = p->next;
    }
    cout << endl;
}
```



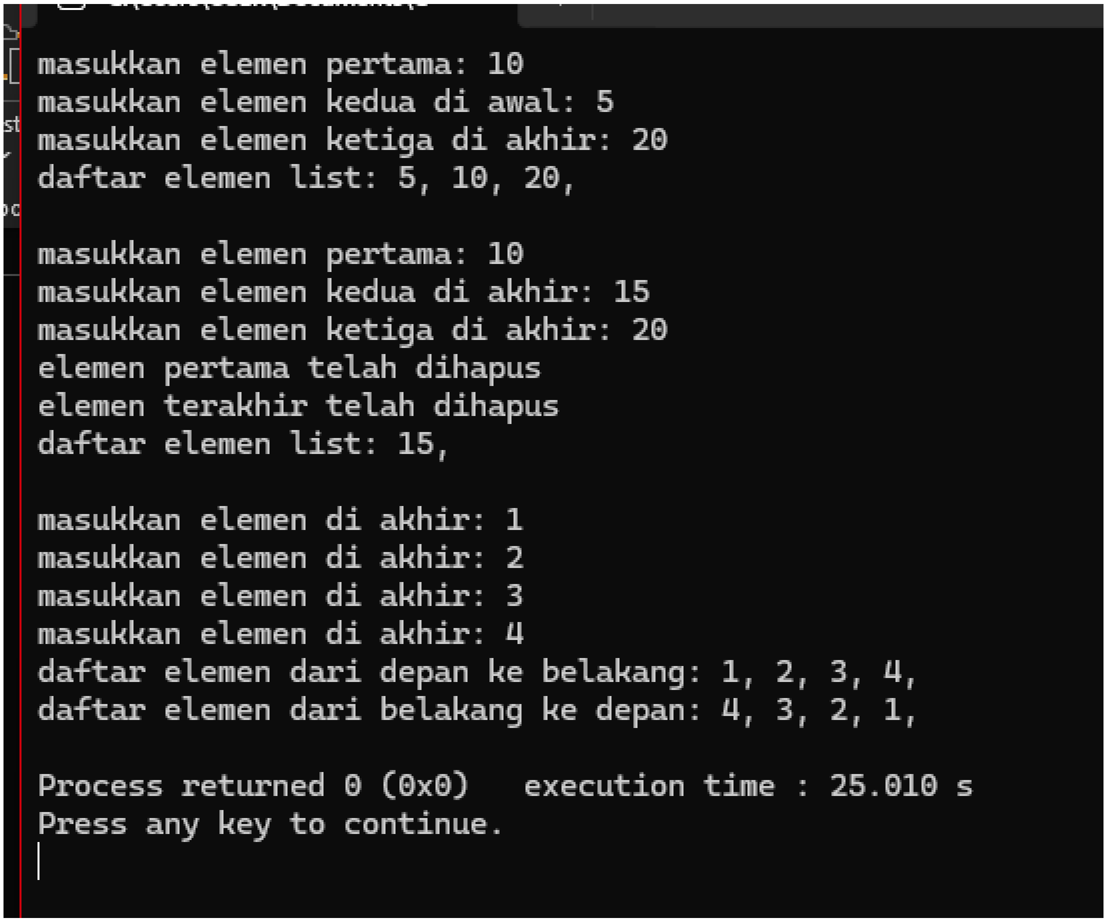
```cpp
#include "DLL.h"

int main() {
    List L;
    createList_103012400153(L);

    infotype x;
    address p, q;

    cout << "masukkan elemen pertama: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    cout << "masukkan elemen kedua di awal: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertFirst_103012400153(L, p);

    cout << "masukkan elemen ketiga di akhir: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    printInfo_103012400153(L);
    cout << endl;
```



```
    cout << "masukkan elemen pertama: ";
    cin >> x;
    p = allocate_103012400153(x);
    createList_103012400153(L); // reset list
    insertLast_103012400153(L, p);

    cout << "masukkan elemen kedua di akhir: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    cout << "masukkan elemen ketiga di akhir: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    deleteFirst_103012400153(L, q);
    cout << "elemen pertama telah dihapus" << endl;

    deleteLast_103012400153(L, q);
    cout << "elemen terakhir telah dihapus" << endl;

    printInfo_103012400153(L);
    cout << endl;
```



```
    createList_103012400153(L);
    cout << "masukkan elemen di akhir: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    cout << "masukkan elemen di akhir: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    cout << "masukkan elemen di akhir: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    cout << "masukkan elemen di akhir: ";
    cin >> x;
    p = allocate_103012400153(x);
    insertLast_103012400153(L, p);

    printInfoFrontToBack_103012400153(L);
    printInfoBackToFront_103012400153(L);

    return 0;
}
```



```
masukkan elemen pertama: 10
masukkan elemen kedua di awal: 5
masukkan elemen ketiga di akhir: 20
daftar elemen list: 5, 10, 20,

masukkan elemen pertama: 10
masukkan elemen kedua di akhir: 15
masukkan elemen ketiga di akhir: 20
elemen pertama telah dihapus
elemen terakhir telah dihapus
daftar elemen list: 15,

masukkan elemen di akhir: 1
masukkan elemen di akhir: 2
masukkan elemen di akhir: 3
masukkan elemen di akhir: 4
daftar elemen dari depan ke belakang: 1, 2, 3, 4,
daftar elemen dari belakang ke depan: 4, 3, 2, 1,

Process returned 0 (0x0)   execution time : 25.010 s
Press any key to continue.
```