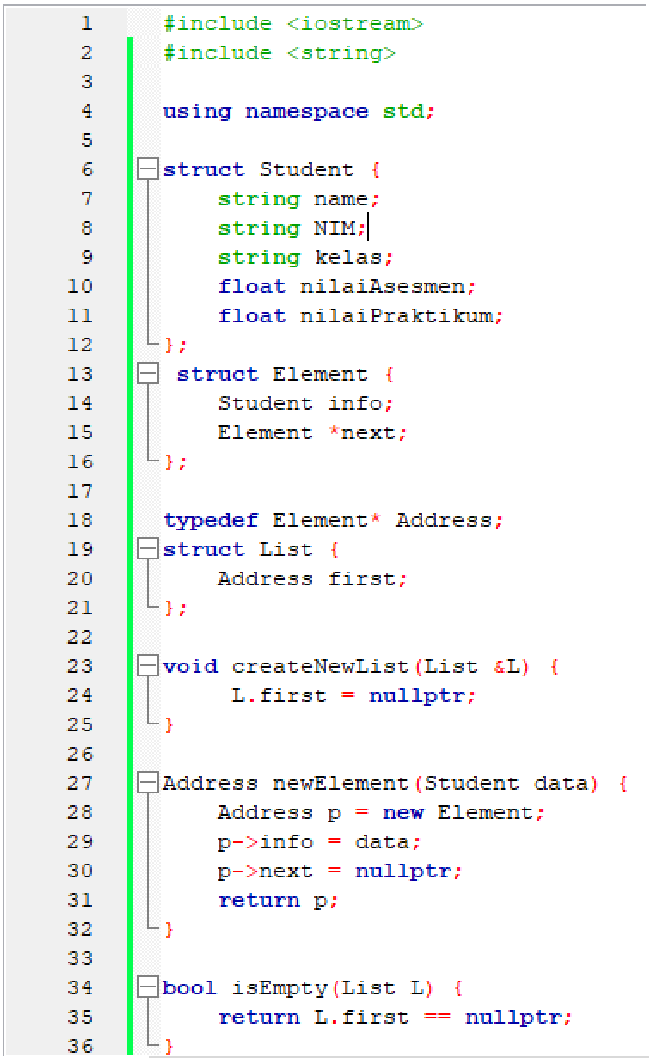
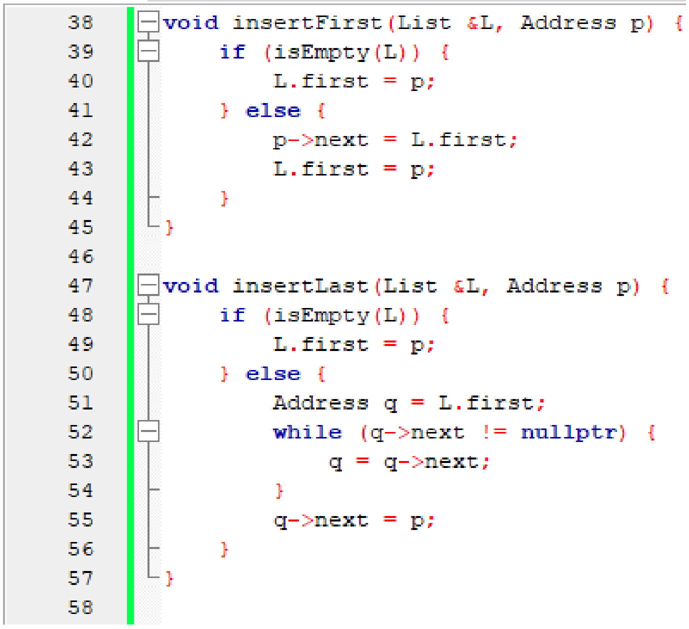
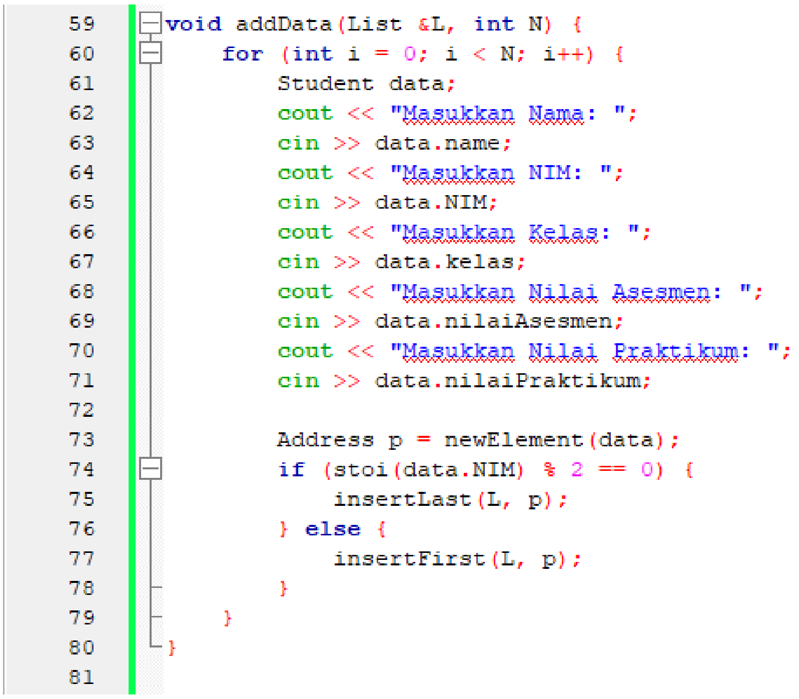


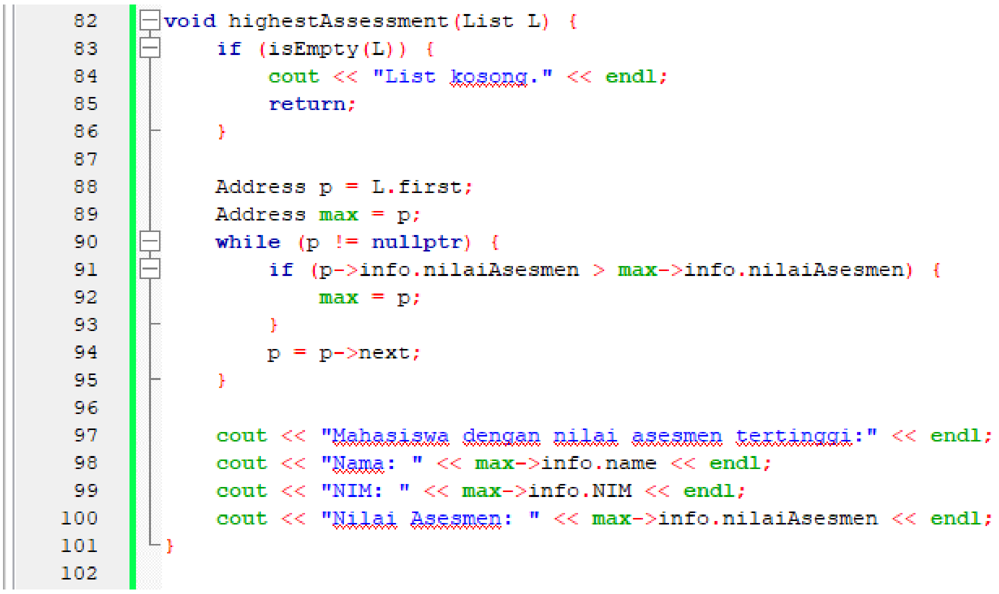
```
#include <iostream>
#include <string>

using namespace std;

struct Student {
    string name;
    string NIM;
    string kelas;
    float nilaiAsesmen;
    float nilaiPraktikum;
};
 struct Element {
    Student info;
    Element *next;
};

typedef Element* Address;
struct List {
    Address first;
};

void createNewList(List &L) {
     L.first = nullptr;
}

Address newElement(Student data) {
    Address p = new Element;
    p->info = data;
    p->next = nullptr;
    return p;
}

bool isEmpty(List L) {
    return L.first == nullptr;
}
```



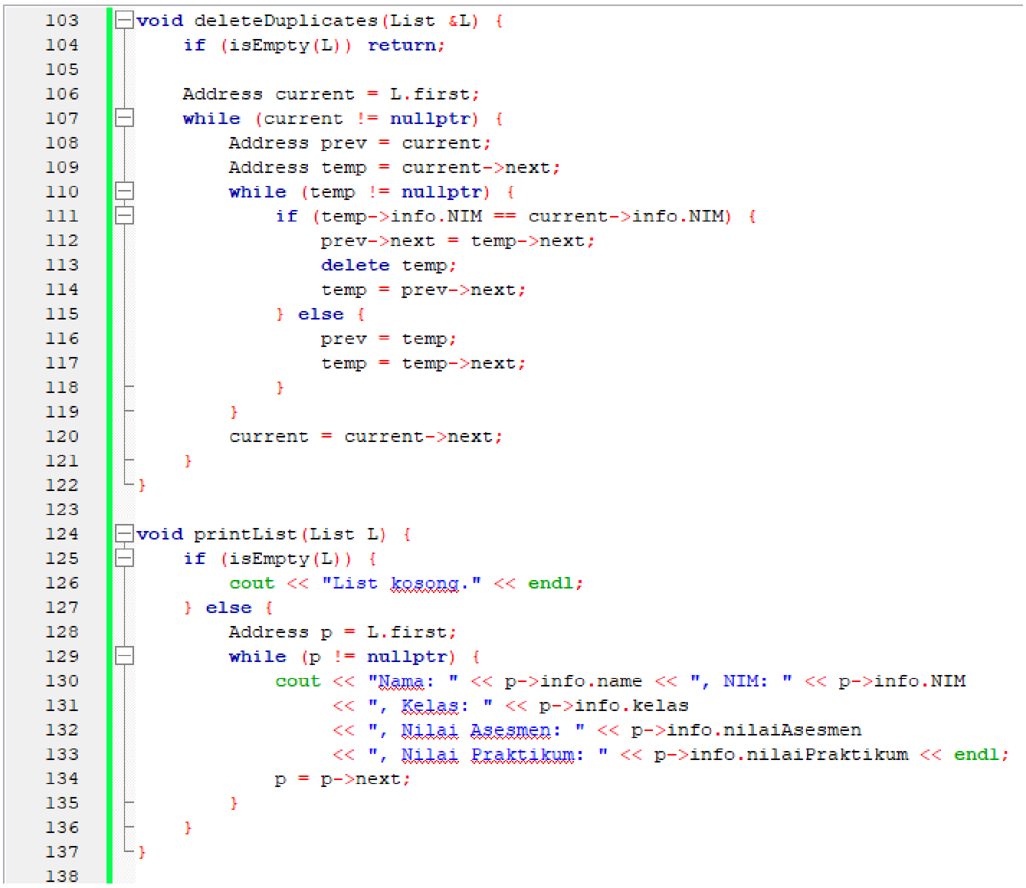
```
void insertFirst(List &L, Address p) {
    if (isEmpty(L)) {
        L.first = p;
    } else {
        p->next = L.first;
        L.first = p;
    }
}

void insertLast(List &L, Address p) {
    if (isEmpty(L)) {
        L.first = p;
    } else {
        Address q = L.first;
        while (q->next != nullptr) {
            q = q->next;
        }
        q->next = p;
    }
}

```



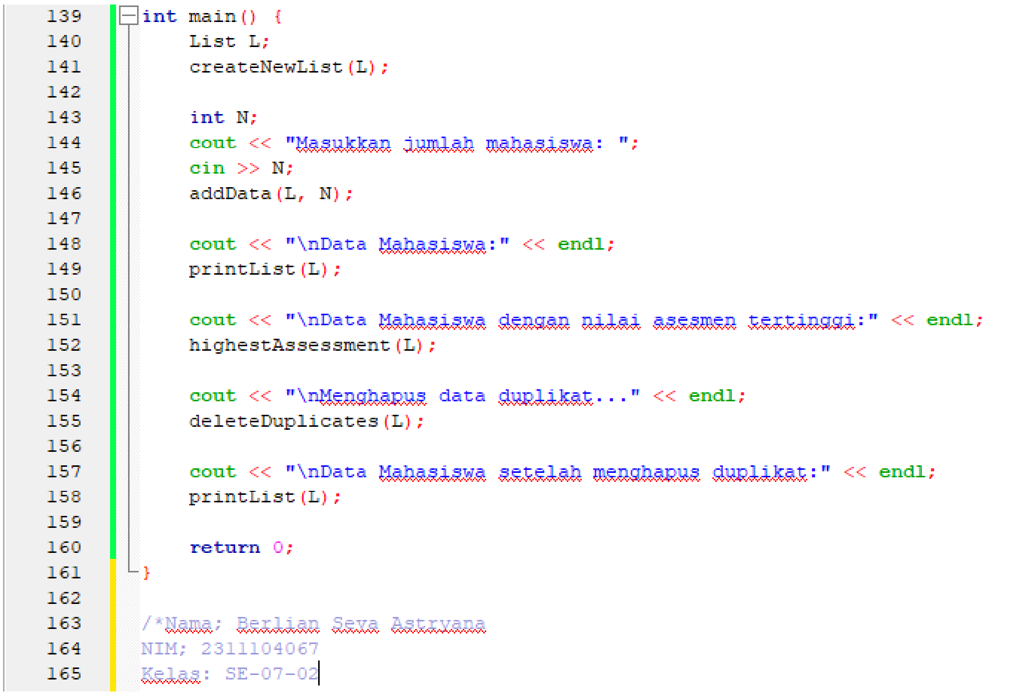
```
void addData(List &L, int N) {
    for (int i = 0; i < N; i++) {
        Student data;
        cout << "Masukkan Nama: ";
        cin >> data.name;
        cout << "Masukkan NIM: ";
        cin >> data.NIM;
        cout << "Masukkan Kelas: ";
        cin >> data.kelas;
        cout << "Masukkan Nilai Asesmen: ";
        cin >> data.nilaiAsesmen;
        cout << "Masukkan Nilai Praktikum: ";
        cin >> data.nilaiPraktikum;

        Address p = newElement(data);
        if (stoi(data.NIM) % 2 == 0) {
            insertLast(L, p);
        } else {
            insertFirst(L, p);
        }
    }
}

```



```
void highestAssessment(List L) {
    if (isEmpty(L)) {
        cout << "List kosong." << endl;
        return;
    }

    Address p = L.first;
    Address max = p;
    while (p != nullptr) {
        if (p->info.nilaiAsesmen > max->info.nilaiAsesmen) {
            max = p;
        }
        p = p->next;
    }

    cout << "Mahasiswa dengan nilai asesmen tertinggi:" << endl;
    cout << "Nama: " << max->info.name << endl;
    cout << "NIM: " << max->info.NIM << endl;
    cout << "Nilai Asesmen: " << max->info.nilaiAsesmen << endl;
}

```

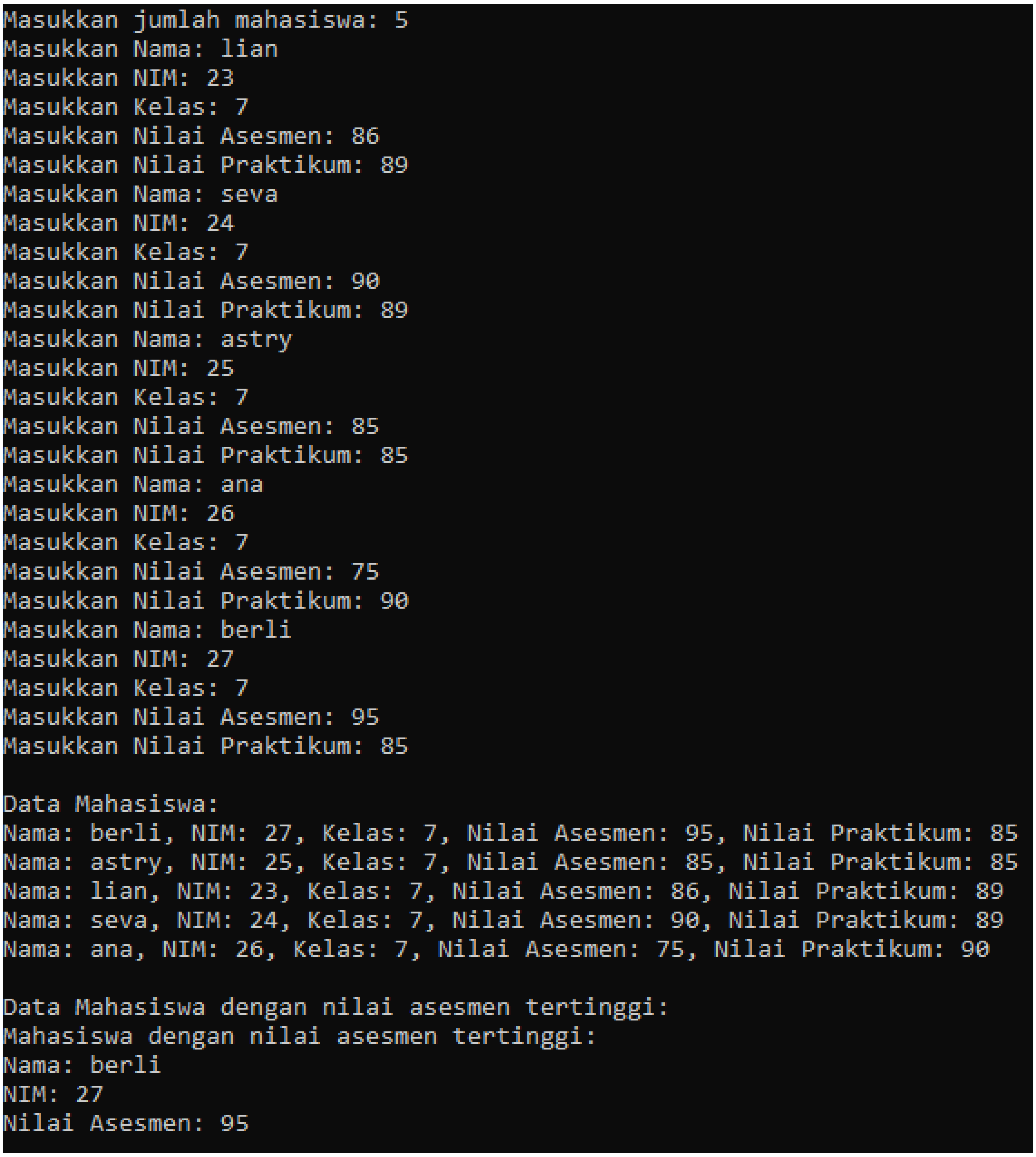


```cpp
void deleteDuplicates(List &L) {
    if (isEmpty(L)) return;

    Address current = L.first;
    while (current != nullptr) {
        Address prev = current;
        Address temp = current->next;
        while (temp != nullptr) {
            if (temp->info.NIM == current->info.NIM) {
                prev->next = temp->next;
                delete temp;
                temp = prev->next;
            } else {
                prev = temp;
                temp = temp->next;
            }
        }
        current = current->next;
    }
}

void printList(List L) {
    if (isEmpty(L)) {
        cout << "List kosong." << endl;
    } else {
        Address p = L.first;
        while (p != nullptr) {
            cout << "Nama: " << p->info.name << ", NIM: " << p->info.NIM
                 << ", Kelas: " << p->info.kelas
                 << ", Nilai Asesmen: " << p->info.nilaiAsesmen
                 << ", Nilai Praktikum: " << p->info.nilaiPraktikum << endl;
            p = p->next;
        }
    }
}

```

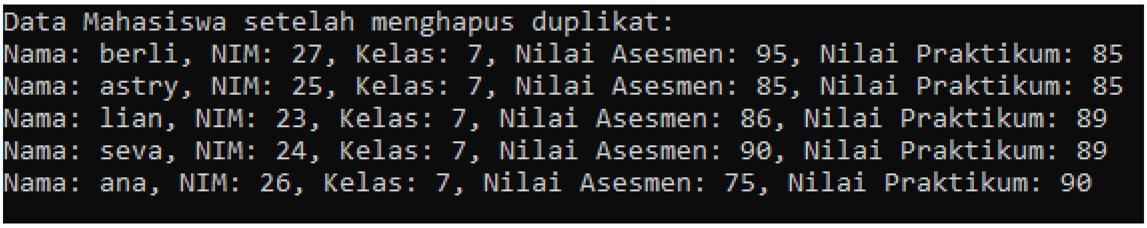


```cpp
int main() {
    List L;
    createNewList(L);

    int N;
    cout << "Masukkan jumlah mahasiswa: ";
    cin >> N;
    addData(L, N);

    cout << "\nData Mahasiswa:" << endl;
    printList(L);

    cout << "\nData Mahasiswa dengan nilai asesmen tertinggi:" << endl;
    highestAssessment(L);

    cout << "\nMenghapus data duplikat..." << endl;
    deleteDuplicates(L);

    cout << "\nData Mahasiswa setelah menghapus duplikat:" << endl;
    printList(L);

    return 0;
}

/*Nama; Berlian Seva Astryana
NIM; 2311104067
Kelas: SE-07-02
```

Output:



```
Masukkan jumlah mahasiswa: 5
Masukkan Nama: lian
Masukkan NIM: 23
Masukkan Kelas: 7
Masukkan Nilai Asesmen: 86
Masukkan Nilai Praktikum: 89
Masukkan Nama: seva
Masukkan NIM: 24
Masukkan Kelas: 7
Masukkan Nilai Asesmen: 90
Masukkan Nilai Praktikum: 89
Masukkan Nama: astry
Masukkan NIM: 25
Masukkan Kelas: 7
Masukkan Nilai Asesmen: 85
Masukkan Nilai Praktikum: 85
Masukkan Nama: ana
Masukkan NIM: 26
Masukkan Kelas: 7
Masukkan Nilai Asesmen: 75
Masukkan Nilai Praktikum: 90
Masukkan Nama: berli
Masukkan NIM: 27
Masukkan Kelas: 7
Masukkan Nilai Asesmen: 95
Masukkan Nilai Praktikum: 85

Data Mahasiswa:
Nama: berli, NIM: 27, Kelas: 7, Nilai Asesmen: 95, Nilai Praktikum: 85
Nama: astry, NIM: 25, Kelas: 7, Nilai Asesmen: 85, Nilai Praktikum: 85
Nama: lian, NIM: 23, Kelas: 7, Nilai Asesmen: 86, Nilai Praktikum: 89
Nama: seva, NIM: 24, Kelas: 7, Nilai Asesmen: 90, Nilai Praktikum: 89
Nama: ana, NIM: 26, Kelas: 7, Nilai Asesmen: 75, Nilai Praktikum: 90

Data Mahasiswa dengan nilai asesmen tertinggi:
Mahasiswa dengan nilai asesmen tertinggi:
Nama: berli
NIM: 27
Nilai Asesmen: 95
```



```
Data Mahasiswa setelah menghapus duplikat:
Nama: berli, NIM: 27, Kelas: 7, Nilai Asesmen: 95, Nilai Praktikum: 85
Nama: astry, NIM: 25, Kelas: 7, Nilai Asesmen: 85, Nilai Praktikum: 85
Nama: lian, NIM: 23, Kelas: 7, Nilai Asesmen: 86, Nilai Praktikum: 89
Nama: seva, NIM: 24, Kelas: 7, Nilai Asesmen: 90, Nilai Praktikum: 89
Nama: ana, NIM: 26, Kelas: 7, Nilai Asesmen: 75, Nilai Praktikum: 90
```