#include <iostream>

using namespace std;

struct Node {

int data;

Node\* prev;

Node\* next;

};

// Fungsi untuk menyisipkan simpul di awal linked list.

void sisipkanSimpulAwal(Node\*& head, Node\*& tail, int data) {

Node\* simpulBaru = new Node();

simpulBaru->data = data;

if (head == nullptr) {

simpulBaru->next = nullptr;

simpulBaru->prev = nullptr;

head = simpulBaru;

tail = simpulBaru;

}

else {

simpulBaru->next = head;

head->prev = simpulBaru;

head = simpulBaru;

}

}

// Fungsi untuk menampilkan linked list dengan tampilan LIFO.

void tampilkanLIFO(Node\* head) {

Node\* temp = head;

cout << "Tampilan LIFO : ";

while (temp != nullptr) {

cout << temp->data << " ";

temp = temp->next;

}

cout << endl;

}

// Fungsi untuk menampilkan linked list dengan tampilan FIFO.

void tampilkanFIFO(Node\* tail) {

Node\* temp = tail;

cout << "Tampilan FIFO : ";

while (temp != nullptr) {

cout << temp->data << " ";

temp = temp->prev;

}

cout << endl;

}

// Fungsi untuk mencari sebuah simpul tertentu.

void cariSimpul(Node\* head, int data) {

Node\* temp = head;

int posisi = 1;

if (head == nullptr) {

cout << "List belum dibuat" << endl;

}

else {

while (temp != nullptr && temp->data != data) {

temp = temp->next;

posisi++;

}

if (temp != nullptr) cout << "data " << data << " ditemukan di node ke-" << posisi << endl;

else cout << "data " << data << " tidak ditemukan" << endl;

}

}

// Fungsi untuk menghapus simpul tertentu.

void hapusSimpul(Node\*& head, Node\*& tail, int data) {

Node\* temp = head;

while (temp != nullptr && temp->data != data) {

temp = temp->next;

}

if (temp == nullptr) return; // Simpul tidak ditemukan

if (temp == head) {

head = head->next;

if (head != nullptr) head->prev = nullptr;

else tail = nullptr;

}

else if (temp == tail) {

tail = tail->prev;

tail->next = nullptr;

}

else {

temp->prev->next = temp->next;

temp->next->prev = temp->prev;

}

delete temp;

}

// Fungsi untuk menghapus seluruh linked list.

void hapusLinkedList(Node\*& head) {

while (head != nullptr) {

Node\* temp = head;

head = head->next;

delete temp;

}

}

int main() {

Node\* head = nullptr;

Node\* tail = nullptr;

sisipkanSimpulAwal(head, tail, 10);

sisipkanSimpulAwal(head, tail, 20);

sisipkanSimpulAwal(head, tail, 30);

sisipkanSimpulAwal(head, tail, 40);

sisipkanSimpulAwal(head, tail, 50);

sisipkanSimpulAwal(head, tail, 60);

tampilkanLIFO(head);

tampilkanFIFO(tail);

int choice, data;

do {

cout << "\nMenu:\n";

cout << "1. Cari simpul\n";

cout << "2. Hapus simpul\n";

cout << "3. Hapus seluruh linked list\n";

cout << "4. Keluar\n";

cout << "Pilihan Anda: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Masukkan nilai simpul yang ingin dicari: ";

cin >> data;

cariSimpul(head, data);

break;

case 2:

cout << "Masukkan nilai simpul yang ingin dihapus: ";

cin >> data;

hapusSimpul(head, tail, data);

break;

case 3:

hapusLinkedList(head);

cout << "Seluruh linked list berhasil dihapus." << endl;

break;

case 4:

cout << "Terima kasih!" << endl;

break;

default:

cout << "Pilihan tidak valid. Silakan coba lagi." << endl;

}

} while (choice != 4);

return 0;

}

