#include <iostream>

struct Node {

int key;

Node\* next;

};

Node\* createNode(int key) {

Node\* newNode = new Node;

newNode->key = key;

newNode->next = nullptr;

return newNode;

}

void printList(Node\* head) {

Node\* current = head;

while (current != nullptr) {

std::cout << current->key << " ";

current = current->next;

}

std::cout << std::endl;

}

void insertNode(Node\*& head, int position, int key) {

Node\* newNode = createNode(key);

if (position == 1) {

newNode->next = head;

head = newNode;

return;

}

Node\* current = head;

for (int i = 1; i < position - 1 && current != nullptr; ++i) {

current = current->next;

}

if (current == nullptr) {

std::cout << "Ошибка: Невозможно добавить элемент на указанную позицию." << std::endl;

delete newNode;

return;

}

newNode->next = current->next;

current->next = newNode;

}

void deleteNode(Node\*& head, int position) {

if (head == nullptr) {

std::cout << "Ошибка: Список пуст." << std::endl;

return;

}

Node\* temp = head;

if (position == 1) {

head = temp->next;

delete temp;

return;

}

for (int i = 1; temp != nullptr && i < position - 1; ++i) {

temp = temp->next;

}

if (temp == nullptr || temp->next == nullptr) {

std::cout << "Ошибка: Невозможно удалить элемент с указанным номером." << std::endl;

return;

}

Node\* nodeToDelete = temp->next;

temp->next = nodeToDelete->next;

delete nodeToDelete;

}

int main() {

Node\* head = nullptr;

// Добавление элементов в список

for (int i = 1; i <= 5; ++i) {

insertNode(head, i, i \* 10);

}

std::cout << "Исходный список: ";

printList(head);

// Удаление элемента с номером 3

deleteNode(head, 3);

std::cout << "Список после удаления элемента с номером 3: ";

printList(head);

// Добавление элемента со значением 75 на позицию 2

insertNode(head, 2, 75);

std::cout << "Список после добавления элемента со значением 75 на позицию 2: ";

printList(head);

// Освобождение памяти

while (head != nullptr) {

Node\* temp = head;

head = head->next;

delete temp;

}

return 0;

}