Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема : "Двунаправленный список"

Выполнила работу:

Студентка группы РИС-22-1Б

Верхоланцева Е.С.

Проверила:

Доцент кафедры ИТАС

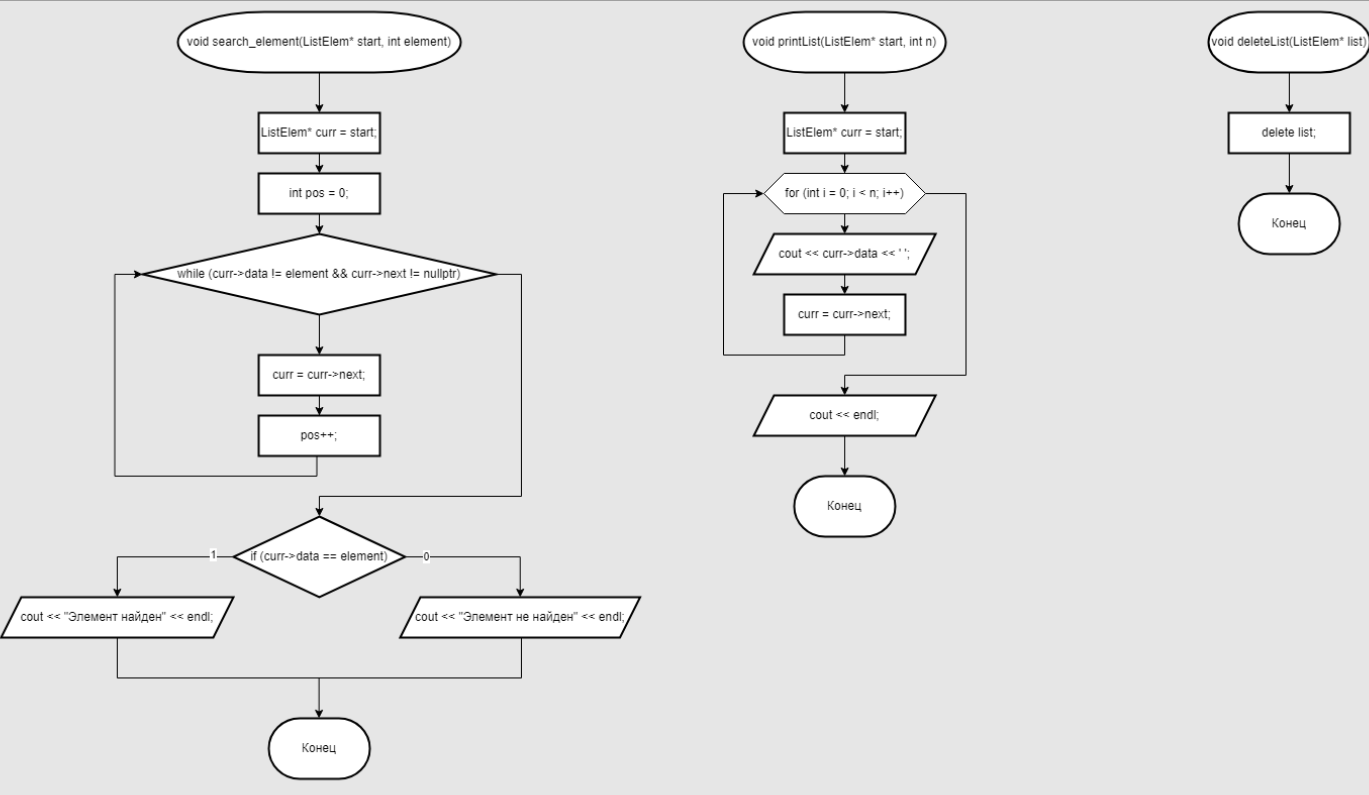
Полякова О.А.

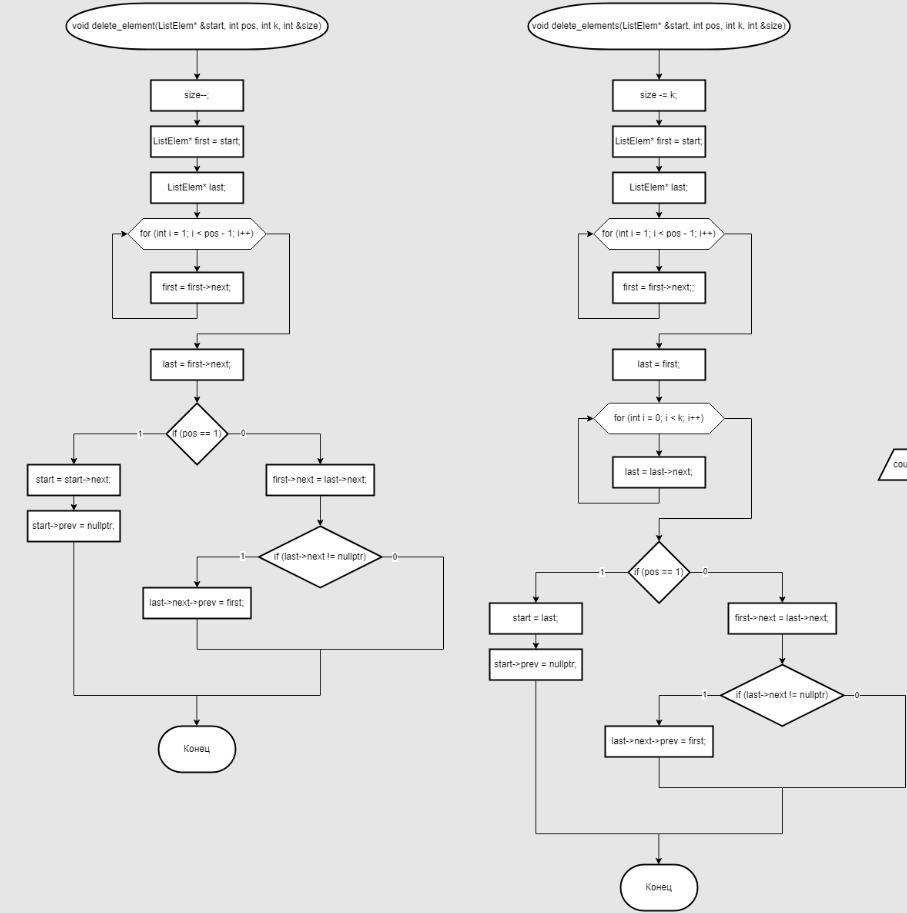
Г. Пермь-2023

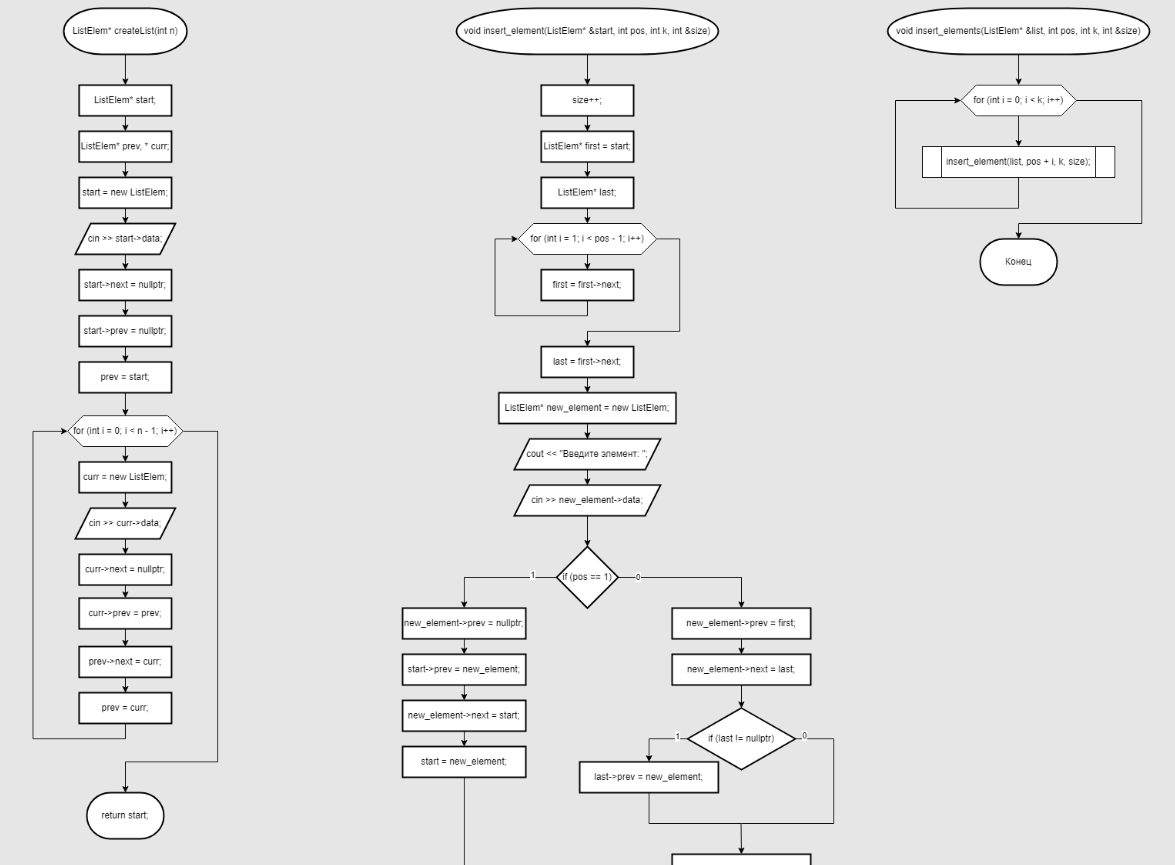
**1. Постановка задачи.**

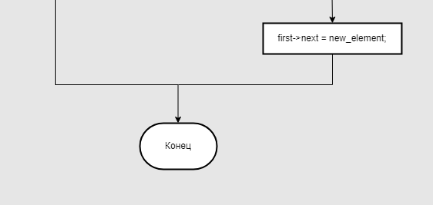
Создать двунаправленный список, добавить в него k элементов, удалить k элементов, реализовать поиск элемента в списке по его значению.

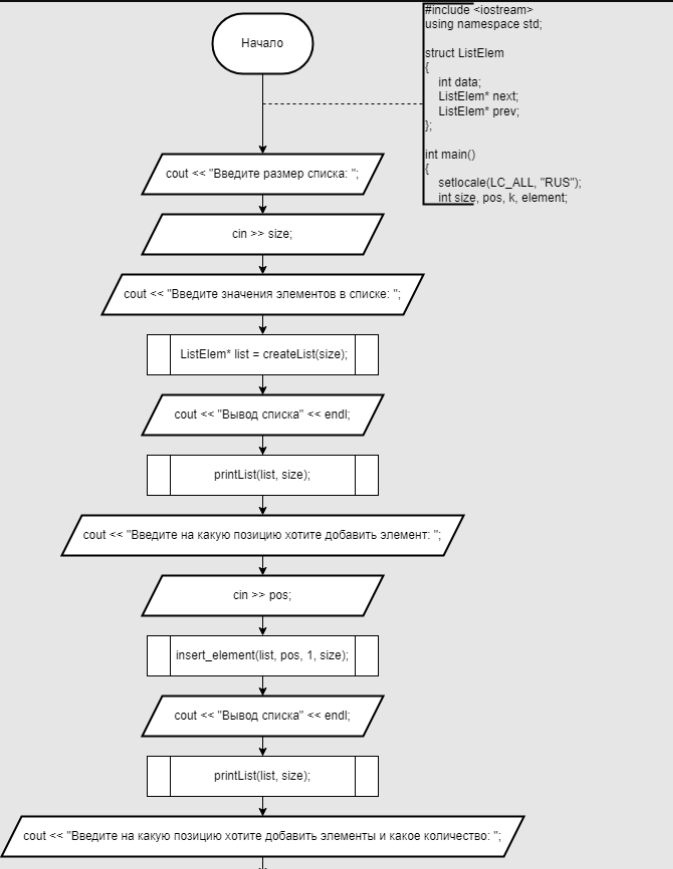
**2. Блок-схема.**

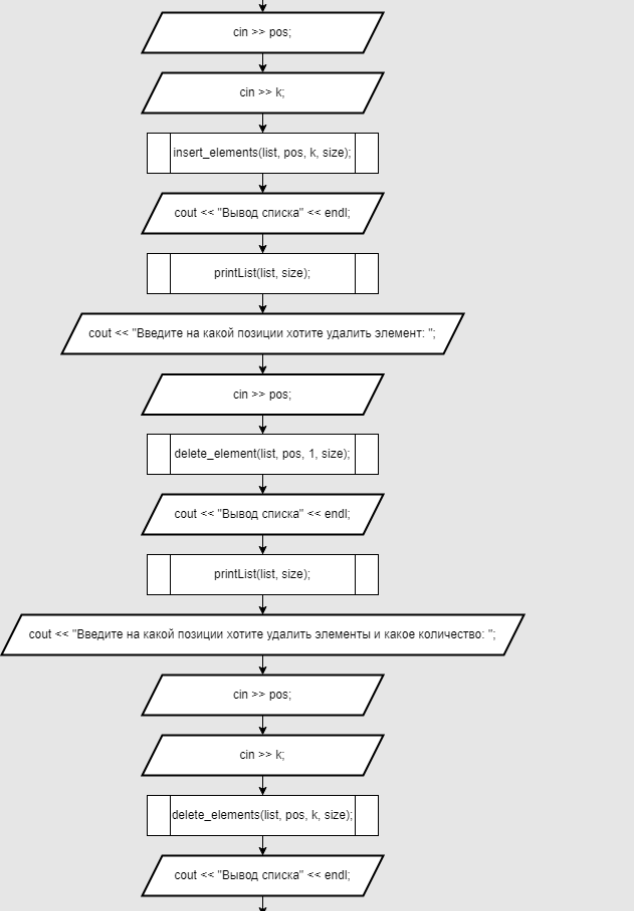


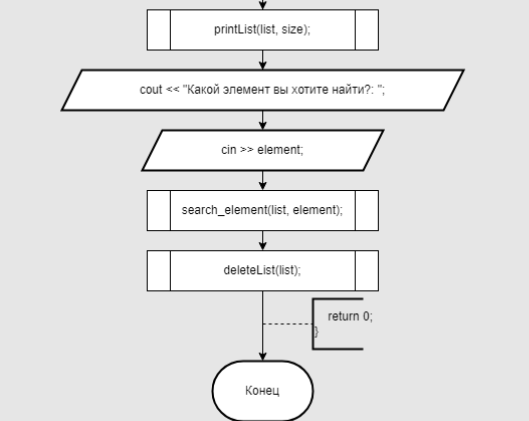












**3. Код.**

#include <iostream>

using namespace std;

// двунаправленный список

struct ListElem // элемент списка

{

int data;

ListElem\* next;

ListElem\* prev;

};

ListElem\* createList(int n) // создание двунаправленного списка

{

ListElem\* start;

ListElem\* prev, \* curr;

start = new ListElem;

cin >> start->data;

start->next = nullptr;

start->prev = nullptr;

prev = start;

for (int i = 0; i < n - 1; i++)

{

curr = new ListElem;

cin >> curr->data;

curr->next = nullptr;

curr->prev = prev;

prev->next = curr;

prev = curr;

}

return start;

}

void insert\_element(ListElem\*& start, int pos, int k, int& size) // вставка одного элемента на заданную позицию

{

size++;

ListElem\* first = start;

ListElem\* last;

for (int i = 1; i < pos - 1; i++)

{

first = first->next;

}

last = first->next;

ListElem\* new\_element = new ListElem;

cout << "Введите элемент: ";

cin >> new\_element->data;

if (pos == 1)

{

new\_element->prev = nullptr;

start->prev = new\_element;

new\_element->next = start;

start = new\_element;

}

else

{

new\_element->prev = first;

new\_element->next = last;

if (last != nullptr)

{

last->prev = new\_element;

}

first->next = new\_element;

}

}

void insert\_elements(ListElem\*& list, int pos, int k, int& size) // вставка k элементов на заданную позицию

{

for (int i = 0; i < k; i++)

{

insert\_element(list, pos + i, k, size);

}

}

void delete\_element(ListElem\*& start, int pos, int k, int& size) // удаление одного элемента с заданной позиции

{

size--;

ListElem\* first = start;

ListElem\* last;

for (int i = 1; i < pos - 1; i++)

{

first = first->next;

}

last = first->next;

if (pos == 1)

{

start = start->next;

start->prev = nullptr;

}

else

{

first->next = last->next;

if (last->next != nullptr)

{

last->next->prev = first;

}

}

}

void delete\_elements(ListElem\*& start, int pos, int k, int& size) // удаление k элементов с заданной позиции

{

size -= k;

ListElem\* first = start;

ListElem\* last;

for (int i = 1; i < pos - 1; i++)

{

first = first->next;;

}

last = first;

for (int i = 0; i < k; i++)

{

last = last->next;

}

if (pos == 1)

{

start = last;

start->prev = nullptr;

}

else

{

first->next = last->next;

if (last->next != nullptr)

{

last->next->prev = first;

}

}

}

void search\_element(ListElem\* start, int element) // поиск элемента в списке

{

ListElem\* curr = start;

int pos = 0;

while (curr->data != element && curr->next != nullptr)

{

curr = curr->next;

pos++;

}

if (curr->data == element)

{

cout << "Элемент найден." << endl;

}

else

{

cout << "Элемент не найден." << endl;

}

}

void printList(ListElem\* start, int n) // вывод элементов списка

{

ListElem\* curr = start;

for (int i = 0; i < n; i++)

{

cout << curr->data << ' ';

curr = curr->next;

}

cout << endl;

}

void deleteList(ListElem\* list) // удаление списка

{

delete list;

}

int main()

{

setlocale(LC\_ALL, "RUS");

int size, pos, k, element;

cout << "Введите размер списка: ";

cin >> size;

cout << "Введите значения элементов в списке: ";

ListElem\* list = createList(size);

cout << "Вывод списка:" << endl;

printList(list, size);

cout << "Введите на какую позицию хотите добавить элемент: ";

cin >> pos;

insert\_element(list, pos, 1, size);

cout << "Вывод списка:" << endl;

printList(list, size);

cout << "Введите на какую позицию хотите добавить элементы и какое количество: ";

cin >> pos;

cin >> k;

insert\_elements(list, pos, k, size);

cout << "Вывод списка:" << endl;

printList(list, size);

cout << "Введите на какой позиции хотите удалить элемент: ";

cin >> pos;

delete\_element(list, pos, 1, size);

cout << "Вывод списка:" << endl;

printList(list, size);

cout << "Введите на какой позиции хотите удалить элементы и какое количество: ";

cin >> pos;

cin >> k;

delete\_elements(list, pos, k, size);

cout << "Вывод списка:" << endl;

printList(list, size);

cout << "Какой элемент вы хотите найти?: ";

cin >> element;

search\_element(list, element);

deleteList(list);

return 0;

}

**4. Скриншот вывода.**

