Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**О Т Ч Е Т**

**по лабораторной работе №11.2 (Двусвязные списки)**

**по дисциплине**

**«Основы алгоритмизации и программирования» семестр 2**

Выполнил студент гр. ИВТ-21-1б

Ипатов Дмитрий Сергеевич

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Д.В.

(оценка) (подпись)

г. Пермь-2022

**Постановка задачи:**

1. Сформировать двусвязный список. Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

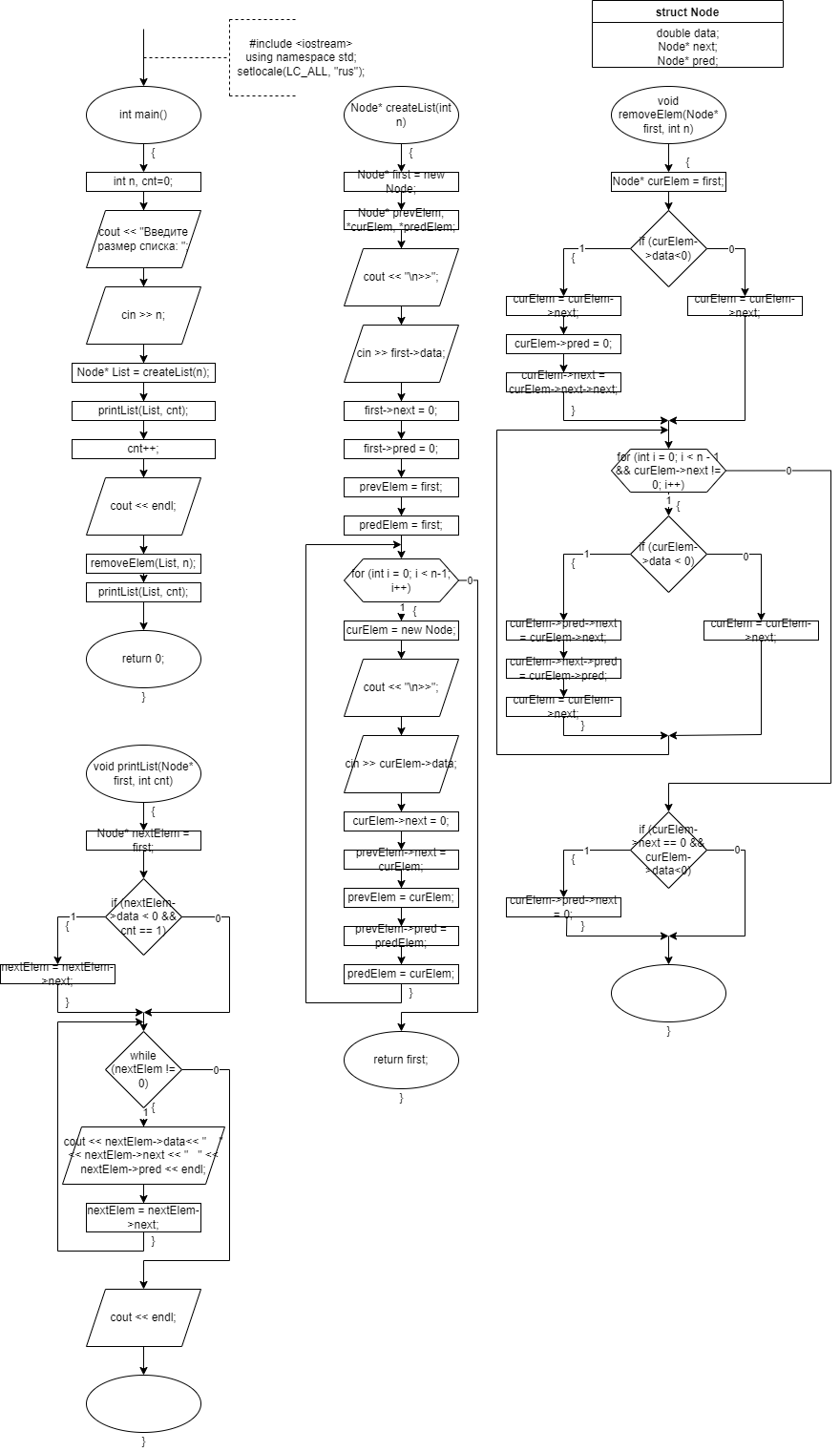
5. Удалить соответствующую структуру из памяти.

Тип информационного поля double. Удалить из списка все элементы с отрицательными информационными полями.

Анализ задачи:

1. Создаем структуру Node с информационным полем double data и с указателем типа структуры на следующий и предыдущий элементы;
2. Создаем список через отдельную функцию, в которую передаем количество элементов, создаем указатель на начало списка и два вспомогательных, задаем первый узел, заходим в цикл и создаем следующие узлы, присваивая в их адресные поля адрес на следующий и предыдущий элемент от каждого;
3. В другой функции мы будем проверять элементы списка условие меньше ли информационное поле нуля если да, то меняем указатели и удаляем элемент, если нет, то переходим на следующий элемент:
4. Выводим список.

**Блок схема:**

****

**Код программы:**

#include <iostream>

using namespace std;

struct Node

{

double data;

Node\* next;

Node\* pred;

};

Node\* createList(int n)

{

Node\* first = new Node;

Node\* prevElem, \*curElem, \*predElem;

cout << "\n>>";

cin >> first->data;

first->next = 0;

first->pred = 0;

prevElem = first;

predElem = first;

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

{

curElem = new Node;

cout << "\n>>";

cin >> curElem->data;

curElem->next = 0;

prevElem->next = curElem;

prevElem = curElem;

prevElem->pred = predElem;

predElem = curElem;

}

return first;

}

void printList(Node\* first, int cnt)

{

Node\* nextElem = first;

if (nextElem->data < 0 && cnt == 1)

{

nextElem = nextElem->next;

}

while (nextElem != 0)

{

cout << nextElem->data<< " " << nextElem->next << " " << nextElem->pred << endl;

nextElem = nextElem->next;

}

cout << endl;

}

void removeElem(Node\* first, int n)

{

Node\* curElem = first;

if (curElem->data<0)

{

curElem = curElem->next;

curElem->pred = 0;

curElem->next = curElem->next->next;

}

else

{

curElem = curElem->next;

}

for (int i = 0; i < n - 1 && curElem->next != 0; i++)

{

if (curElem->data < 0)

{

curElem->pred->next = curElem->next;

curElem->next->pred = curElem->pred;

curElem = curElem->next;

}

else

{

curElem = curElem->next;

}

}

if (curElem->next == 0 && curElem->data<0)

{

curElem->pred->next = 0;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

int n, cnt=0;

cout << "Введите размер списка: ";

cin >> n;

Node\* List = createList(n);

printList(List, cnt);

cnt++;

cout << endl;

removeElem(List, n);

printList(List, cnt);

return 0;

}