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»

**Отчет по Лабораторной работе Однонаправленный список**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Ахунов Руслан Булатович

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

г. Пермь - 2022

**Постановка задачи:**

Вариант 2

Тип информационного поля double. Удалить из списка все элементы с четными номерами (2, 4, 6 и. т. д.).

**Анализ задачи:**

struct Node

{

    double data;

    Node\* ptr\_next = nullptr;

};

struct List

{

    Node\* headnode = nullptr;

    Node\* tailnode = nullptr;

};

Структура Ноды, в которой есть поле для какого-либо значения и указатель на следующую Ноду.

Структура Список, есть указатель типа Нода на начало списка и на его конец.

void pushBack(List& list, int& data)

{

    Node\* newnode = new Node;

    newnode->data = data;

    if (list.headnode == nullptr)

    {

        list.headnode = newnode;

        list.tailnode = newnode;

    }

    else

    {

        list.tailnode->ptr\_next = newnode;

        list.tailnode = newnode;

    }

}

Функция pushBack, в которую принимается по ссылке список и то значение, которое нужно добавить.

В этой функции мы создаем временную Ноду и записываем в неё значение переменной, затем проверяем, ни пуст ли список, и если он пуст, то тогда

это Нода становиться как началом, так и концом списка. Если же список не пуст, то к указателю на следующий символ конца списка присваиваем текущую Ноду, но при этом она становиться концом этого списка.

void printList(List& list)

{

    Node\* currentnode = list.headnode;

    while (currentnode != nullptr)

    {

        cout << currentnode->data << " ";

        currentnode = currentnode->ptr\_next;

    }

}

Функция printList, принимаем по ссылке список. Инициализируем временный указатель типа Нода и присваиваем ей значение начала списка. Затем в цикле проверяем, не равен ли он «пустоте», если нет, то выводим в консоль значения data и переходим в нем на следующую Ноду. И так до тех пор, пока не закончиться список.

void changeList(List& list)

{

    Node\* currentnode = list.headnode;

    Node\* prednode = list.headnode;

    int count = 1;

    while (currentnode != nullptr)

    {

        if (count % 2 == 0)

        {

            prednode->ptr\_next = currentnode->ptr\_next;

            prednode = currentnode->ptr\_next;

        }

        currentnode = currentnode->ptr\_next;

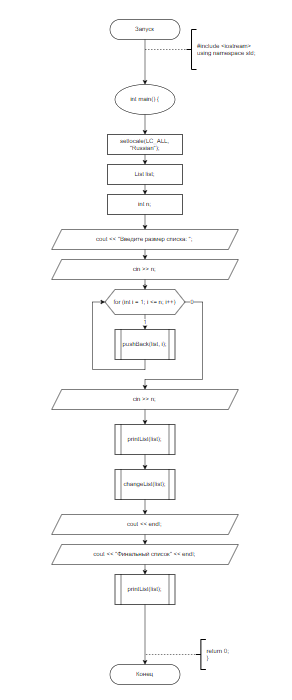
        count++;

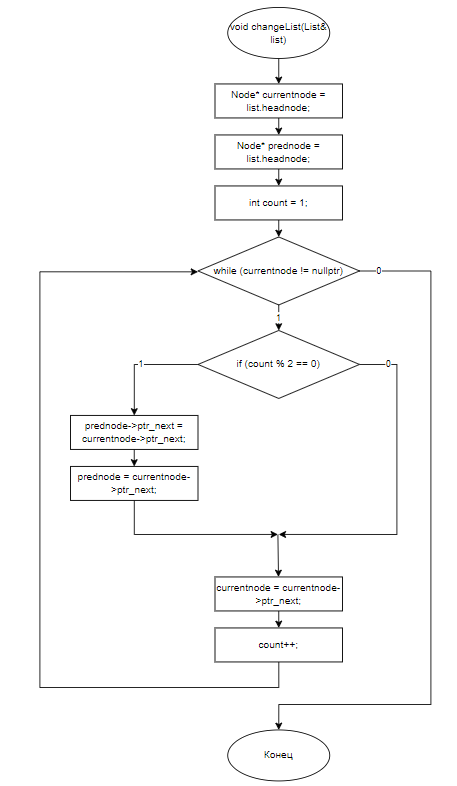
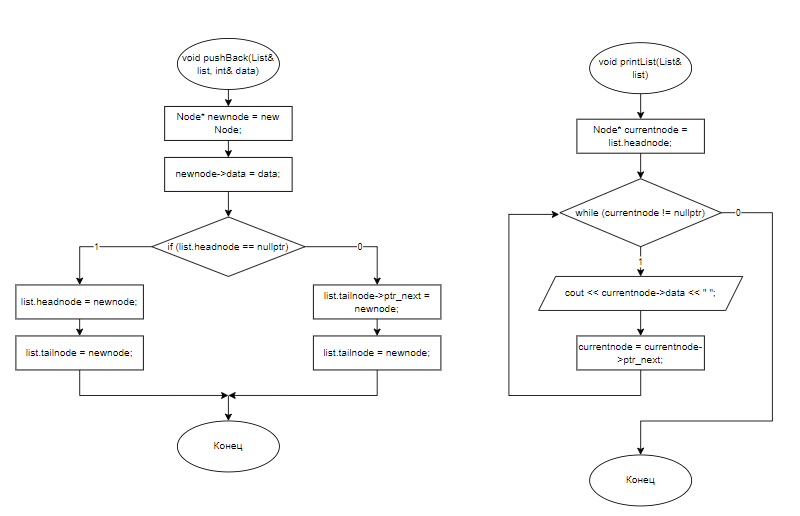
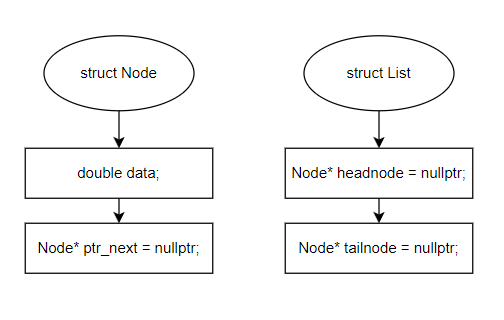
    }

}

Так как мне надо удалить из списка все элементы с четными номерами, то я создаю переменную для порядкового номера Ноды. И если эта переменная четная, то я заменяю указатель на следующую Ноду у это предыдущей Ноды, на указатель на следующую Ноды текущей Ноды.

**Блок-схема**

****

****

**Код программы**

#include <iostream>

using namespace std;

struct Node

{

    double data;

    Node\* ptr\_next = nullptr;

};

struct List

{

    Node\* headnode = nullptr;

    Node\* tailnode = nullptr;

};

void pushBack(List& list, int& data)

{

    Node\* newnode = new Node;

    newnode->data = data;

    if (list.headnode == nullptr)

    {

        list.headnode = newnode;

        list.tailnode = newnode;

    }

    else

    {

        list.tailnode->ptr\_next = newnode;

        list.tailnode = newnode;

    }

}

void printList(List& list)

{

    Node\* currentnode = list.headnode;

    while (currentnode != nullptr)

    {

        cout << currentnode->data << " ";

        currentnode = currentnode->ptr\_next;

    }

}

void changeList(List& list)

{

    Node\* currentnode = list.headnode;

    Node\* prednode = list.headnode;

    int count = 1;

    while (currentnode != nullptr)

    {

        if (count % 2 == 0)

        {

            prednode->ptr\_next = currentnode->ptr\_next;

            prednode = currentnode->ptr\_next;

        }

        currentnode = currentnode->ptr\_next;

        count++;

    }

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    List list;

    int n;

    cout << "Введите размер списка: ";

    cin >> n;

    for (int i = 1; i <= n; i++)

    {

        pushBack(list, i);

    }

    cout << "Изначальный список" << endl;

    printList(list);

    changeList(list);

    cout << endl;

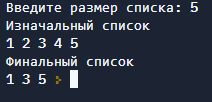
    cout << "Финальный список" << endl;

    printList(list);

    return 0;

}

**Вывод программы**

****