Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Электротехнический факультет Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил студент группы ИВТ21-1Б Лебедев Т.А.

Проверил Яруллин Д.В.

Пермь, 2022

**Постановка задачи:**

1. Сформировать однонаправленный и двунаправленный списки или стек и очередь. 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.

**Анализ решения задачи:**

1.создаём структура с именем node

2.создаем информационное поле

3.создаём указатель, который будет указывать на следующий узел списка

4.создаем функцию, которая будет создавать список

5.указатель на начало списка

6.вспомогательные указатели

7.выделяем память под новый элемент

8.просим ввести число(цифру)

9.очищаем от "мусора" указатель

10.создаём цикл в котором будем создавать другие узлы списка

11.выделяем память под новый элемент списка

12.считываем информацию о новом элементе

13.очищаем указатель от "мусора"

14.возвращаем получившийся список

15.создаем функцию для вывода списка

16.создаём указатель, который будет указывать на первый элемент списка

17.цикл while будет выполняться пока не встретиться последний элемент

выводим информационное поле

18.переходим к новому элементу

19.создаем функцию, которая будет удалять узлы списка с чётным информационным полем

20.создаём указатель, который будет указывать на первый элемент списка

21.если встретился элемент списка с информационным полем, то…

22.в информационное поле нужного элемента записываем информационное поле 23.следующего элемента

24.указатель элемента с четным информационным полем ставим на элемент следующий

25.переходим к следующему элементу

26.возвращаем получившийся список

**Описание переменных:**

n- количество элементов списка

data-информационное поле структуры

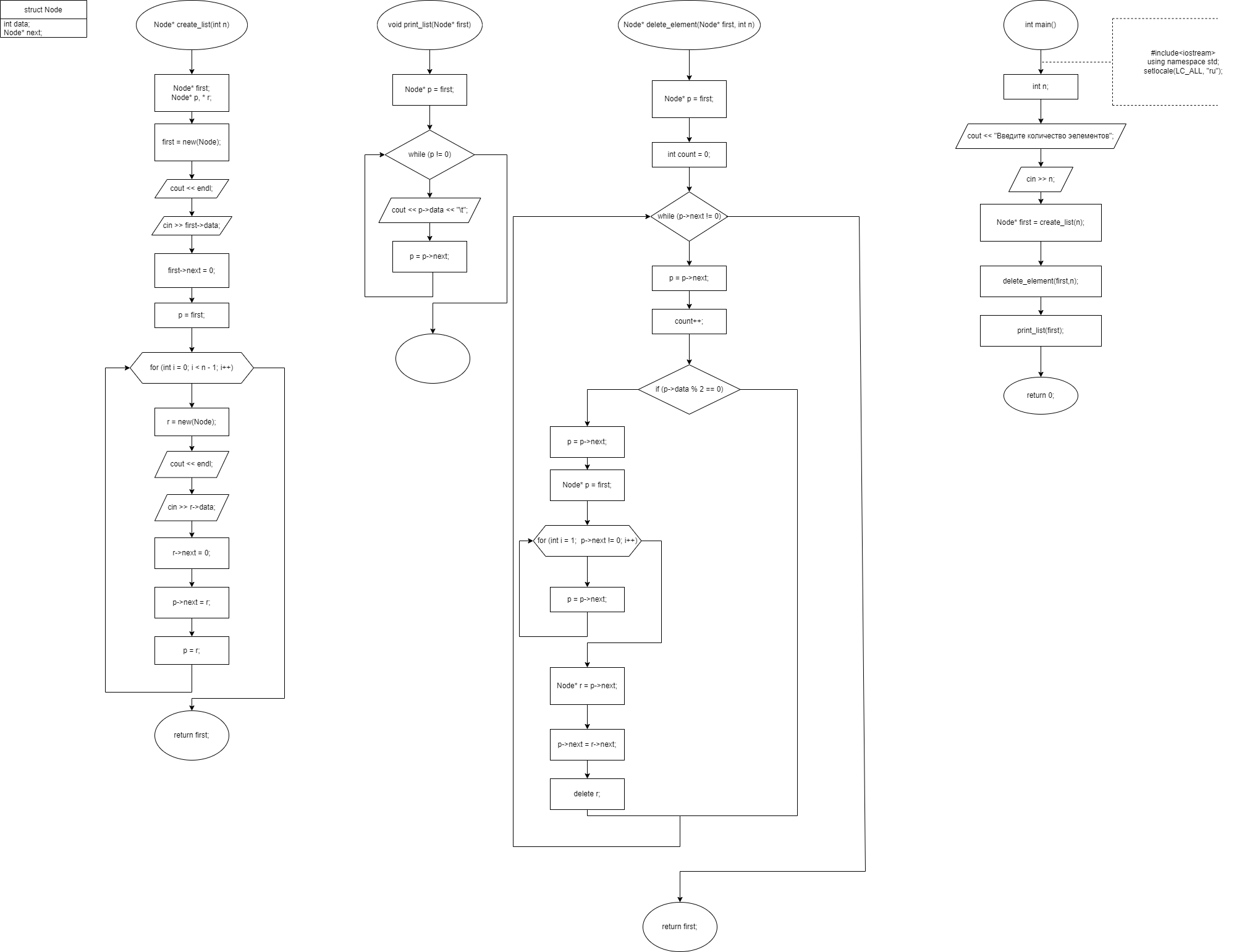
\*next-указатель на следующий элемент

\*pred-указатель на предыдущий элемент

\*first-переменная, в которой будет хранится первый элемент списка

\*p,\*r-вспомогательные переменные

**Блок-схема:**

****

**Код программы:**

#include<iostream>

using namespace std;

struct Node//создаём структура с именем node

{

int data;//создаём информационное поле

Node\* next;//создаём указатель, который будет указывать на следующий узел списка

};

Node\* create\_list(int n)//создаём функцию, которая будет создавать список

{

Node\* first;//указатель на начало списка

Node\* p, \* r;//вспомогательные указатели

first = new(Node);//выделяем память под новый элемент

cout << endl;

cin >> first->data;//просим ввести число(цифру)

first->next = 0;//очищаем от "мусора" указатель

p = first;

for (int i = 0; i < n - 1; i++)//создаём цикл в котором будем создавать другие узлы списка

{

r = new(Node);//выделяем память под новый элемент списка

cout << endl;

cin >> r->data;//считываем информацию о новом элементе

r->next = 0;//очищаем указатель от "мусора"

p->next = r;

p = r;

}

return first;//возвращаем получившийся список

}

void print\_list(Node\* first)//создаём функцию для вывода списка

{

Node\* p = first;//создаём указатель, который будет указывать на первый элемент списка

while (p != 0)// цикл while будет выполняться пока не встретиться последний элемент

{

if (p->data%2 != 0)

{

cout << p->data << "\t";//выводим информационное поле

}

p = p->next;//переходим к новому элементу

}

}

Node\* delete\_element(Node\* first, int n)//создаём функцию, которая будет удалять узлы списка с чётным информационным полем

{

Node\* p = first;//создаём указатель, который будет указывать на первый элемент списка

while (p!= 0)//цикл while будет выполняться пока не встретиться последний элемент

{

if (p->data % 2 == 0 && p->next != 0)//если встретился элемент списка с информационным полем, то...

{

p->data = p->next->data;//в информационное поле нужного элемента записываем информационное поле следующего элемента

p->next = p->next->next;//указатель элемента с чётным информационным полем ставим на элемент следующий

}

p = p->next;//переходим к следующему элементу

}

return first;//возвращаем получившийся список

}

int main()

{

int n;

setlocale(LC\_ALL, "ru");

cout << "Введите количество эелементов";

cin >> n;

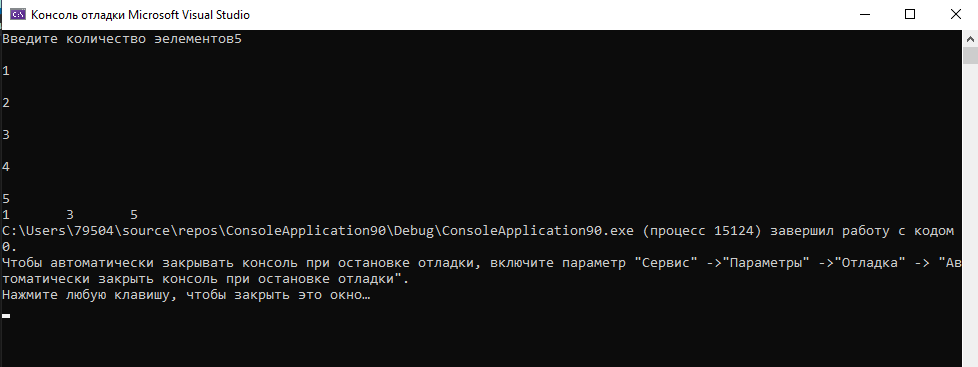
Node\* first = create\_list(n);

delete\_element(first, n);

print\_list(first);

}

**Скриншоты результатов:**

****