

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**Пермский национальный исследовательский
политехнический университет**

Факультет Электротехнический Кафедра ИТАС

ОТЧЁТ

о лабораторной работе №11 (двунаправленный список)

Выполнил:
Студент группы ИВТ-23-1Б
Пискунов Д. А.

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2023

Постановка задач

1. Создание списка.
2. Добавление элемента в список (в соответствии со своим вариантом).
3. Удаление элемента из списка (в соответствии со своим вариантом).
4. Печать списка.
5. Запись списка в файл.
6. Уничтожение списка.
7. Восстановление списка из файла.

Вариант 20

Записи в линейном списке содержат ключевое поле типа `*char` (строка символов). Сформировать двунаправленный список. Удалить элемент с заданным ключом. Добавить по K элементов в начало и в конец списка.

Текст программы

```
1  #include<iostream>
2  #include<string>
3  #include<fstream>
4
5  using namespace std;
6
7  struct Queue {
8      char data;
9      Queue* next = nullptr;
10 };
11 struct que {
12     Queue* front = nullptr;
13     Queue* back = nullptr;
14     int size;
15     void push(char data) {
16         Queue* new_node = new Queue;
17         new_node->data = data;
18         new_node->next = nullptr;
19         if (size == 0) {
20             front = new_node;
21             back = new_node;
22         }
23         else {
24             back->next = new_node;
25             back = new_node;
26         }
27         size++;
28     }
29     char pop() {
30         char datatmp = front->data;
31         Queue* tmp = front;
32         if (size == 1) {
33             back = nullptr;
34         }
35         front = front->next;
36         size--;
37         delete tmp;
38         return datatmp;
39     }
40 };
41
42 void print(que& s) {
43     if (s.size == 0) {
44         cout << "Очередь пуста." << endl;
45         return;
46     }
47     char datatmp;
48     int size = s.size;
49     for (int i = 0; i < size; i++) {
50         datatmp = s.pop();
51         cout << datatmp;
52         s.push(datatmp);
53     }
54 }
55
56 void add(que& s, char data, int n) {
57     que tmp;
58     char datatmp;
59     int size = s.size;
60     for (int i = 0; i < size; i++) {
61         datatmp = s.pop();
```

```

64         datatmp = s.pop();
65         if (i == n - 1) {
66             s.push(data);
67         }
68         s.push(datatmp);
69     }
70 }
71
72 void del(queue< s, int n) {
73     char datatmp;
74     int size = s.size;
75     for (int i = 0; i < size; i++) {
76         datatmp = s.pop();
77         if (i != n - 1) {
78             s.push(datatmp);
79         }
80     }
81 }
82
83 bool search(queue< s, char da) {
84     char datatmp;
85     bool flag = false;
86     int size = s.size;
87     for (int i = 0; i < size; i++) {
88         datatmp = s.pop();
89         if (datatmp == da) {
90             flag = true;
91         }
92         s.push(datatmp);
93     }
94
95     return flag;
96 }
97
98 void saver(queue< s) {
99     ofstream file1("C:\\Users\\MOKASIH\\Desktop\\sem_2\\11\\save.txt");
100     if (file1) {
101         char datatmp;
102         int size = s.size;
103         for (int i = 0; i < size; i++) {
104             datatmp = s.pop();
105             file1 << datatmp;
106             s.push(datatmp);
107         }
108     }
109
110     else {
111         cout << "Файл не открылся";
112     }
113     file1.close();
114 }
115
116 void killer(queue< s) {
117     int size = s.size;
118     for (int i = 0; i < size; i++) {
119         s.pop();
120     }
121 }
122
123 void load(queue< s) {
124     ifstream file1("C:\\Users\\MOKASIH\\Desktop\\sem_2\\11\\save.txt");
125     if (file1) {
126         cin.ignore();
127         string f;

```

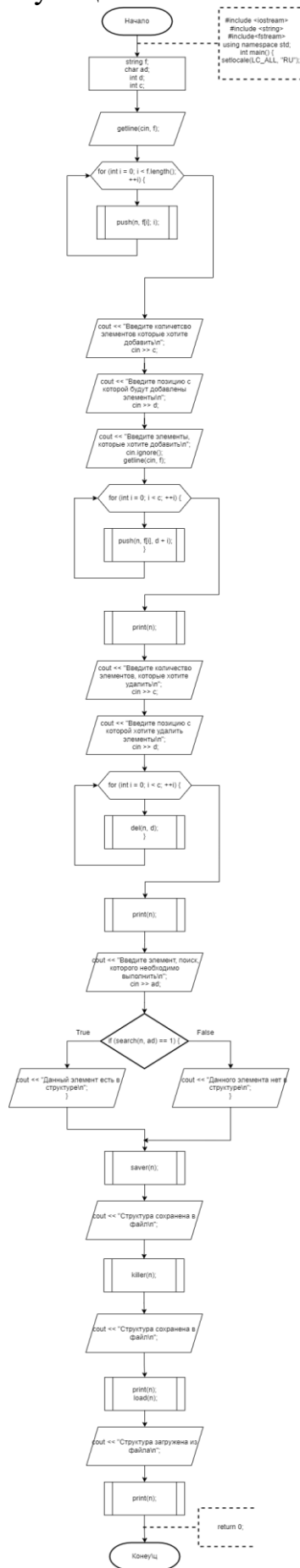
```

127     string f;
128     getline(file1, f);
129     for (int i = 0; i < f.length(); i++) {
130         s.push(f[i]);
131     }
132     file1.close();
133 }
134 else {
135     cout << "Файл не открыт";
136 }
137 }
138
139
140
141
142 int main() {
143     setlocale(LC_ALL, "RU");
144     int p, d;
145     char ad;
146     string f;
147     cout << "Введите символы которыми хотите заполнить очередь\n";
148     getline(cin, f);
149     for (int i = 0; i < f.length(); i++)
150     {
151         s.push(f[i]);
152     }
153
154     cout << "Сформированная очередь\n";
155     print(s);
156
157     cout << "\nВведите количество элементов которые хотите добавить\n";
158     cin >> d;
159     cout << "Введите позиции на которую хотите добавить элементы\n";
160     cin >> p;
161     cout << "Введите элементы которые хотите добавить\n";
162     cin.ignore();
163     getline(cin, f);
164     for (int i = 0; i < d; i++) {
165         add(s, f[i], p + i);
166     }
167     print(s);
168
169     cout << "\nВведите количество элементов которые хотите удалить\n";
170     cin >> d;
171     cout << "Введите позиции с которой хотите удалить элементы\n";
172     cin >> p;
173     for (int i = 0; i < d; i++) {
174         del(s, p);
175     }
176     print(s);
177
178
179     cout << "\nВведите элемент, поиск, которого необходимо выполнить\n";
180     cin >> ad;
181     if (search(s, ad) == 1) {
182         cout << "Данный элемент есть в структуре\n";
183     }
184     else {
185         cout << "Данного элемента нет в структуре\n";
186     }
187
188     saver(s);
189     cout << "Структура сохранена в файл\n";
190     killer(s);
191     cout << "Структура удалена\n";
192     print(s);
193
194     load(s);
195     cout << "Структура загружена из файла\n";
196     print(s);
197
198     return 0;
199 }

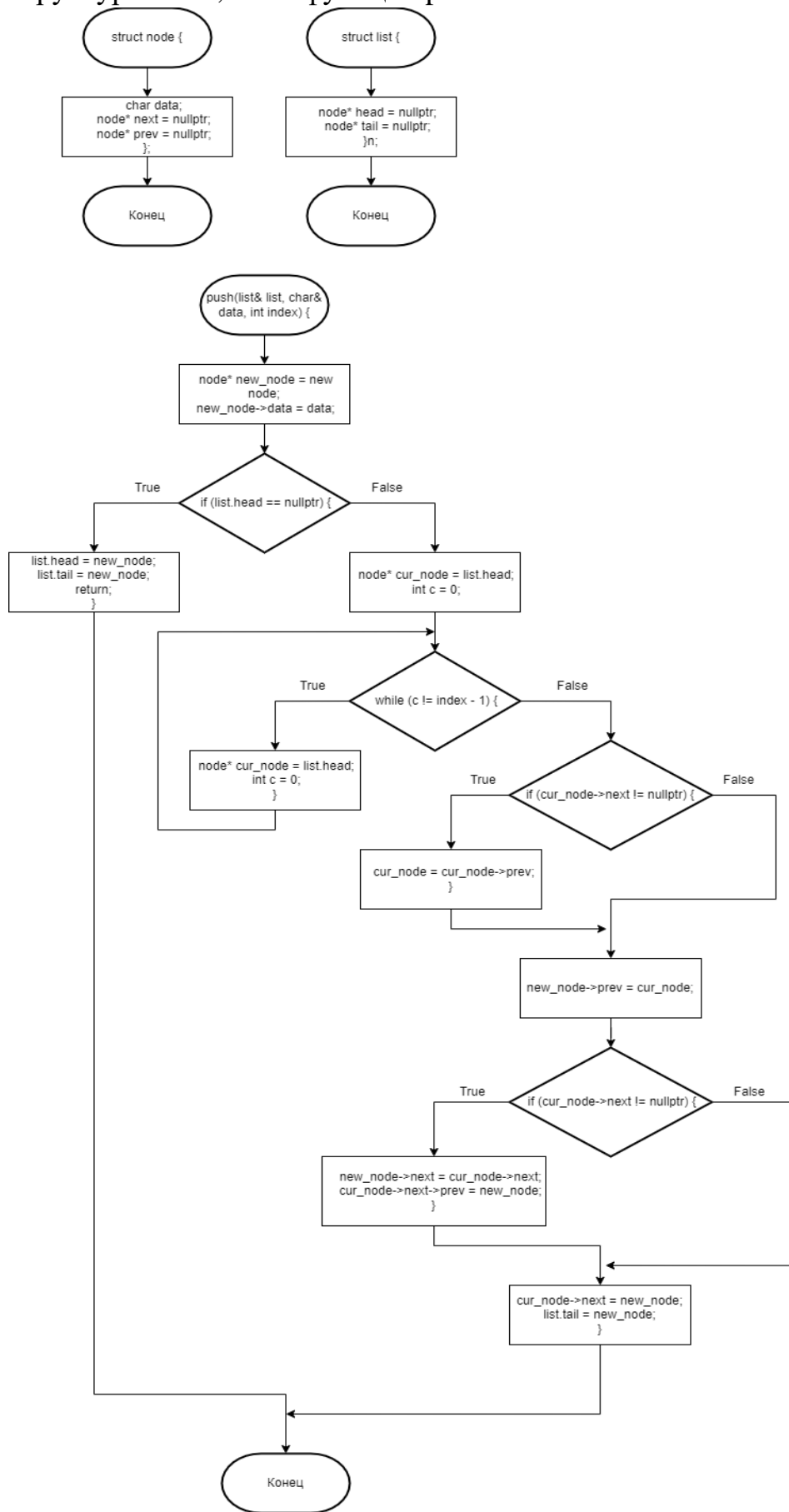
```

3) Блок схема

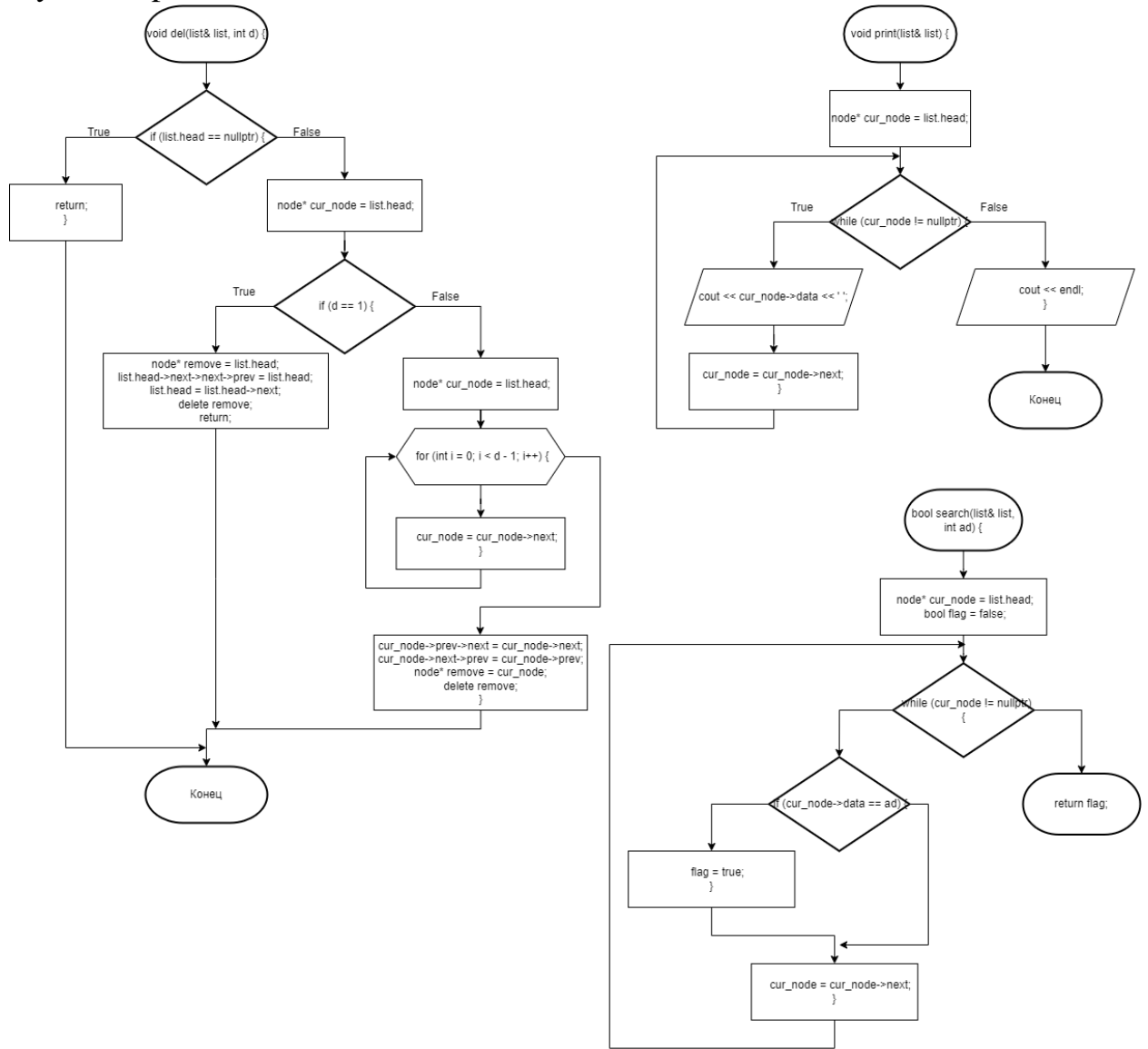
Функция main



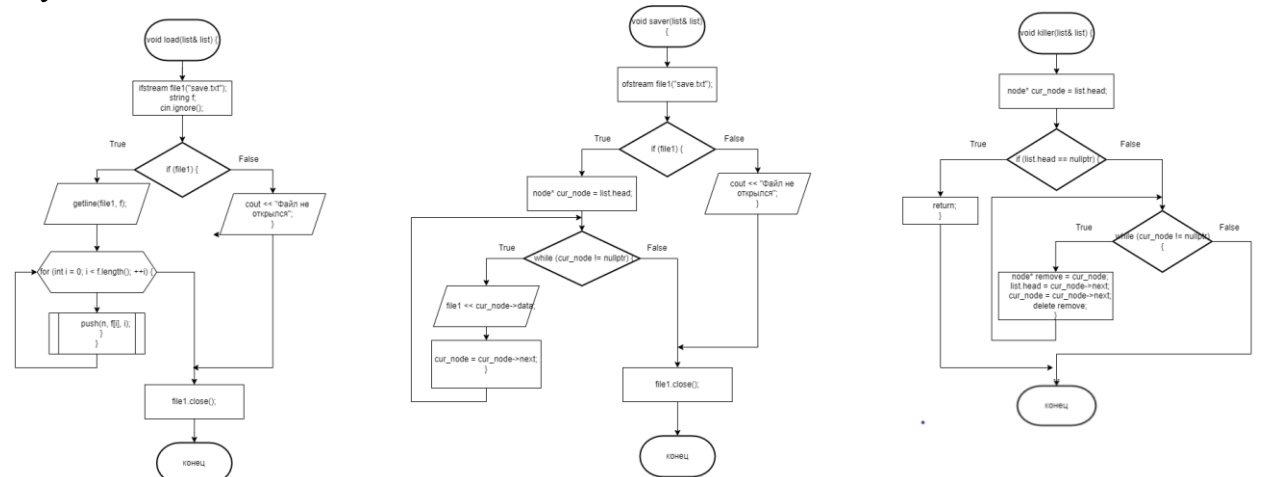
структуры node, list и функция push



Функции print, del, search



Функции saver, killer, load



Тест

```
tractor
Сформированная структура:
t r a s t o r
Введите количество элементов которые хотите добавить
5
Введите позицию на которую хотите добавить элементы
7
Введите элементы которые хотите добавить
e d e t
t r a s t o r   e d e t
Введите количество элементов которые хотите удалить
6
Введите позицию с которой хотите удалить элементы
1
r   e d e t
Введите элемент, поиск, которого необходимо выполнить
r
Данный элемент есть в структуре
Структура сохранена в файл
Структура удалена

Структура загружена из файла
r   e d e t

C:\Users\MOkASiH\Desktop\labs\x64\Debug\labs.exe (процесс 16768) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
```