

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 (однонаправленный 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 однонаправленный 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 using namespace std;
5
6 struct node {
7     char data;
8     node* next = nullptr;
9 };
10
11 struct list {
12     node* head = nullptr;
13     int size = 0;
14 };
15
16 void pushBack(list& list, char& data) {
17     node* new_node = new node;
18     new_node->data = data;
19     if (list.head == nullptr) {
20         list.head = new_node;
21     }
22     else {
23         node* cur_node = list.head;
24         while (cur_node->next != nullptr) {
25             cur_node = cur_node->next;
26         }
27         cur_node->next = new_node;
28     }
29     list.size++;
30 }
31
32 void push(list& list, char& data, int d) {
33     node* pre_node = nullptr;
34     node* cur_node = list.head;
35     node* new_node = new node;
36     new_node->data = data;
37     if (d == 1) {
38         new_node->next = list.head;
39         list.head = new_node;
40         list.size++;
41         return;
42     }
43     for (int i = 0; i < d - 1; i++) {
44         if (i == d - 2) {
45             pre_node = cur_node;
46         }
47         cur_node = cur_node->next;
48     }
49     pre_node->next = new_node;
50     new_node->next = cur_node;
51     list.size++;
52 }
53
54 void print(list& list) {
55
56     void print(list& list) {
57         node* cur_node = list.head;
58         while (cur_node != nullptr) {
59             cout << cur_node->data << ' ';
60             cur_node = cur_node->next;
61         }
62         cout << endl;
63     }
64
65     void del(list& list, int d) {
66         if (list.head == nullptr) {
67             return;
68         }
69         node* cur_node = list.head;
70         if (d == 1) {
71             node* remove = list.head;
72             list.head = list.head->next;
73             delete remove;
74             list.size--;
75             return;
76         }
77         node* pre_node = nullptr;
78         for (int i = 0; i < d - 1; i++) {
79             if (i == d - 2) {
80                 pre_node = cur_node;
81             }
82             cur_node = cur_node->next;
83         }
84         pre_node->next = cur_node->next;
85         node* remove = cur_node;
86         delete remove;
87         list.size--;
88     }
89
90     bool search(list& list, int ad) {
91         node* cur_node = list.head;
92         bool flag = false;
93         while (cur_node != nullptr) {
94             if (cur_node->data == ad) {
95                 flag = true;
96             }
97             cur_node = cur_node->next;
98         }
99         return flag;
100     }
101
102     }
103
104     }
105
106     }
107
108     }
109
110     }
111 }
```

```

109     }
110     return flag;
111 }
112 }
113
114 void saver(list& list) {
115     ofstream file1("C:\\Users\\MOKASIH\\Desktop\\sem_2\\11\\save.txt");
116     if (file1) {
117         node* cur_node = list.head;
118         while (cur_node != nullptr) {
119             file1 << cur_node->data;
120             cur_node = cur_node->next;
121         }
122     }
123     else {
124         cout << "Файл не открылся";
125     }
126     file1.close();
127 }
128
129 void killer(list& list) {
130     node* cur_node = list.head;
131     if (list.head == nullptr) {
132         return;
133     }
134     while (cur_node != nullptr) {
135         node* remove = cur_node;
136         list.head = cur_node->next;
137         cur_node = cur_node->next;
138         delete remove;
139     }
140     list.size = 0;
141 }
142
143 void load(list& list) {
144     ifstream file1("C:\\Users\\MOKASIH\\Desktop\\sem_2\\11\\save.txt");
145     string f;
146     cin.ignore();
147     if (file1) {
148         getline(file1, f);
149         for (int i = 0; i < f.length(); ++i) {
150             pushBack(n, f[i]);
151         }
152     }
153     else {
154         cout << "Файл не открылся";
155     }
156     file1.close();
157 }
158
159
160 int main() {
161     setlocale(LC_ALL, "RU");
162
163     string f;
164     char ad;
165     int d;

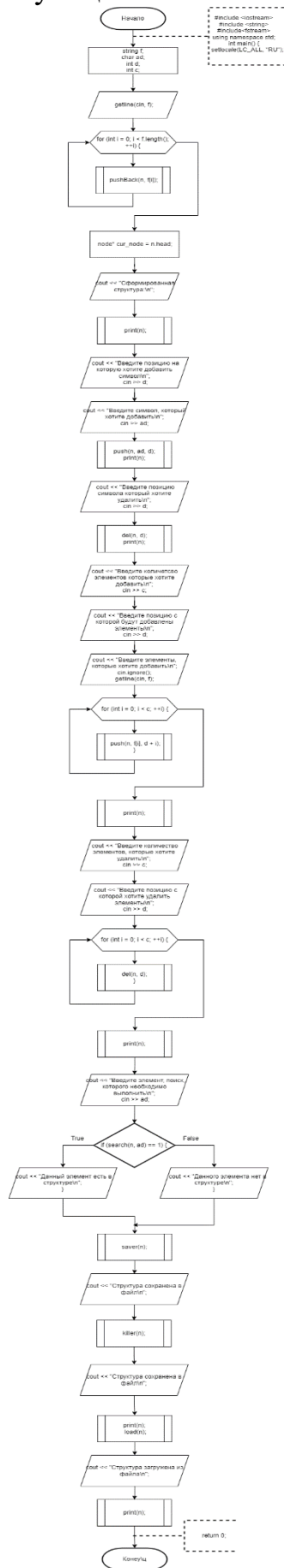
```

```

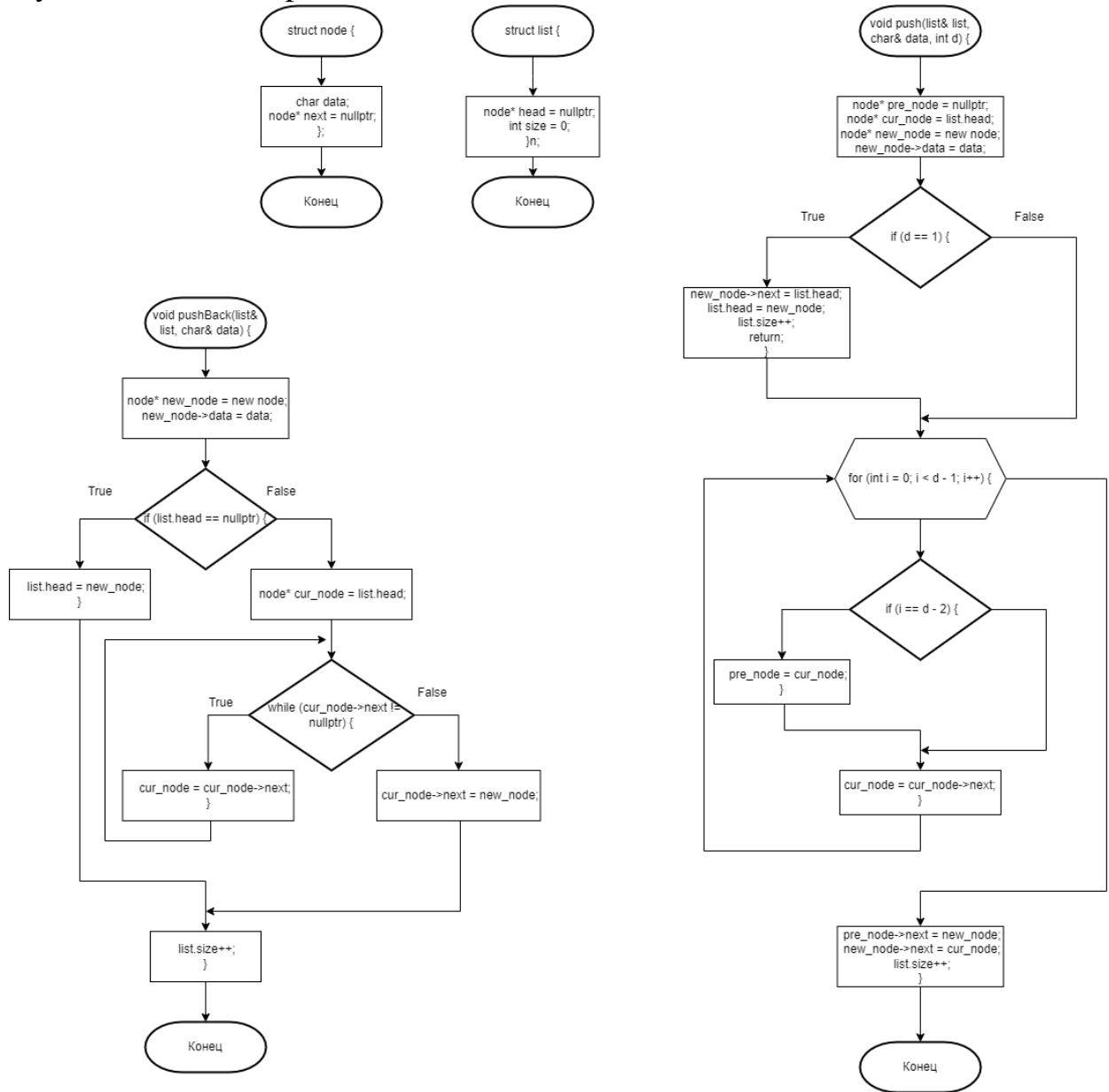
164 char ad;
165 int d;
166 getline(cin, f);
167
168 for (int i = 0; i < f.length(); ++i) {
169     pushBack(n, f[i]);
170 }
171
172 node* cur_node = n.head;
173
174 cout << "Сформированная структура:\n";
175
176 print(n);
177 cout << "Введите позицию на которую хотите добавить символ\n";
178 cin >> d;
179 cout << "Введите символ, который хотите добавить\n";
180 cin >> ad;
181 push(n, ad, d);
182 print(n);
183
184
185 cout << "Введите позицию символа который хотите удалить\n";
186 cin >> d;
187 del(n, d);
188 print(n);
189
190 int c;
191 cout << "Введите количество элементов которые хотите добавить\n";
192 cin >> c;
193 cout << "Введите позицию с которой будут добавлены элементы\n";
194 cin >> d;
195 cout << "Введите элементы, которые хотите добавить\n";
196 cin.ignore();
197 getline(cin, f);
198 for (int i = 0; i < c; ++i) {
199     push(n, f[i], d + i);
200 }
201 print(n);
202
203 cout << "Введите количество элементов, которые хотите удалить\n";
204 cin >> c;
205 cout << "Введите позицию с которой хотите удалить элементы\n";
206 cin >> d;
207 for (int i = 0; i < c; ++i) {
208     del(n, d);
209 }
210 print(n);
211
212 cout << "Введите элемент, поиск, которого необходимо выполнить\n";
213 cin >> ad;
214 if (search(n, ad) == 1) {
215     cout << "Данный элемент есть в структуре\n";
216 }
217 else {
218     cout << "Данного элемента нет в структуре\n";
219 }
220
221 saver(n);
222 cout << "Структура сохранена в файл\n";
223 killer(n);
224 cout << "Структура удалена\n";
225 print(n);
226
227 load(n);
228 cout << "Структура загружена из файла\n";
229 print(n);
230
231 return 0;
232
233 }

```

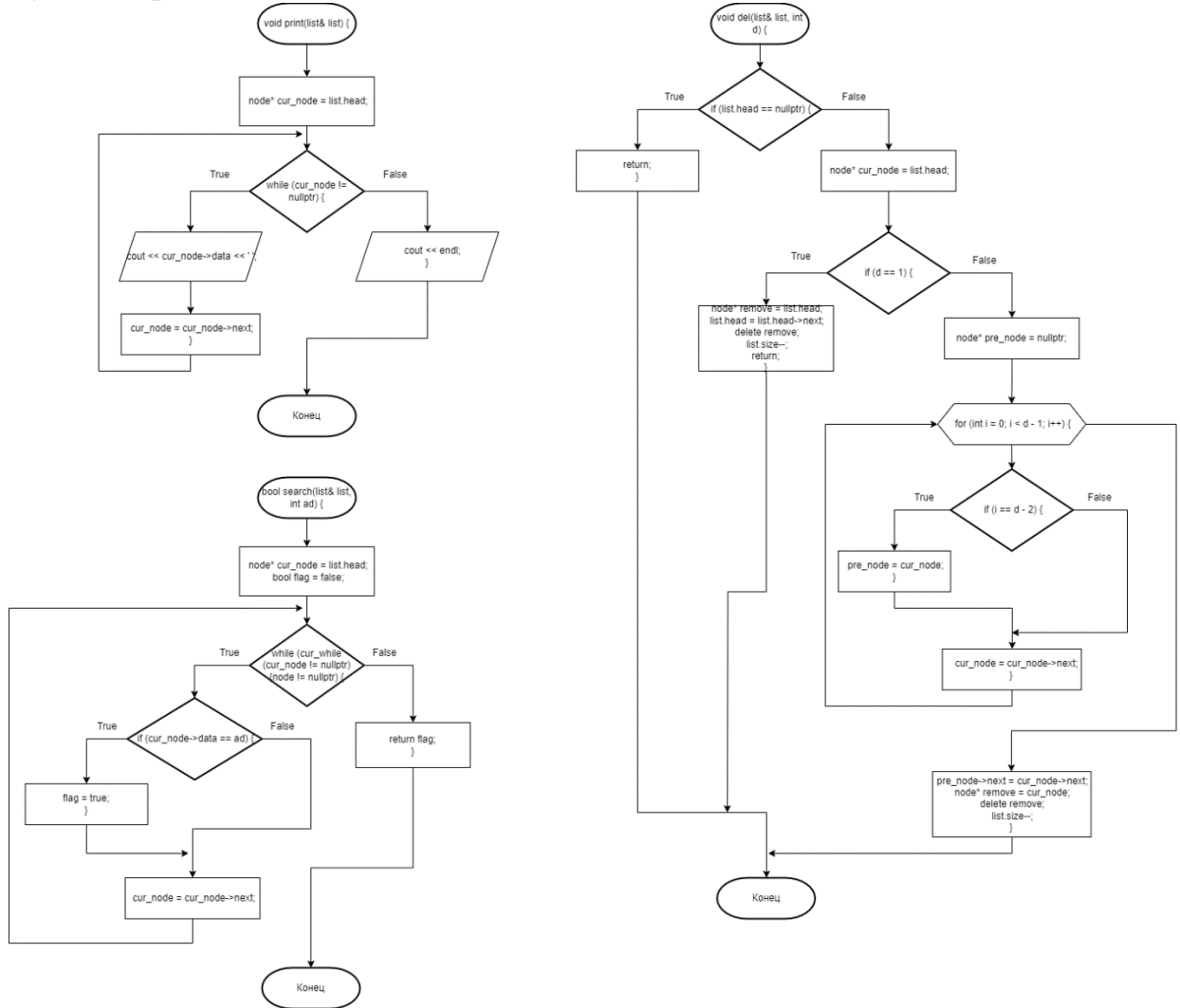
### 3) Блок схема Функция main



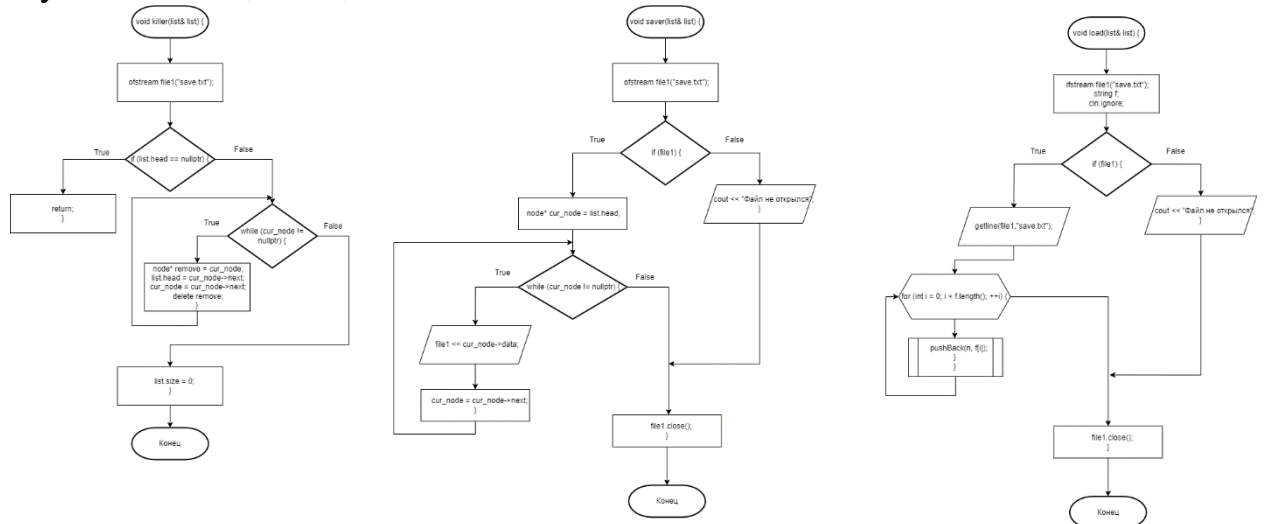
## Функции node, list, pushBack



## Функции print, del, search



## Функции saver, killer, load





## Тест

```
hello
Сформированная структура:
h e l l o
Введите позицию на которую хотите добавить символ
6
Введите символ, который хотите добавить
_
h e l l o _
Введите позицию символа который хотите удалить
6
h e l l o
Введите количество элементов которые хотите добавить
6
Введите позицию с которой будут добавлены элементы
6
Введите элементы, которые хотите добавить
world
h e l l o   w o r l d
Введите количество элементов, которые хотите удалить
6
Введите позицию с которой хотите удалить элементы
1
w o r l d
Введите элемент, поиск, которого необходимо выполнить
o
Данный элемент есть в структуре
Структура сохранена в файл
Структура удалена

Структура загружена из файла
w o r l d

C:\Users\MOkASiH\Desktop\labs\x64\Debug\labs.exe (процесс 10764) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
```