

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 №17

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

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2024

Задача:

Написать программу, выполняющую поиск Бойера-Мура и поиск Кнута-Морриса-Пратта

Текст программы

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  const int CHAR_NUM = 256;
7
8  void calc_char_table(string str, int size, int char_table[CHAR_NUM]) {
9      for (int i = 0; i < CHAR_NUM; i++) {
10         char_table[i] = -1;
11     }
12     for (int i = 0; i < size; i++) {
13         char_table[(int)str[i]] = i;
14     }
15 }
16 void boyer_moore_search(string str, string pattern) {
17     int str_size = str.size();
18     int pattern_size = pattern.size();
19
20     int shift_table[CHAR_NUM];
21
22     calc_char_table(str, pattern_size, shift_table);
23
24     int shift = 0;
25
26     while (shift <= (str_size - pattern_size)) {
27         int i = pattern_size - 1;
28
29         while (i >= 0 and pattern[i] == str[shift + i]) {
30             i--;
31         }
32
33         if (i < 0) {
34             cout << "Паттерн найден на позиции " << shift + 1 << endl;
35             if ((shift + pattern_size) < str_size) {
36                 shift += pattern_size - shift_table[(int)str[shift + pattern_size]];
37             }
38             else {
39                 shift++;
40             }
41         }
42         else {
43             shift += max(1, i - shift_table[(int)str[shift + i]]);
44         }
45     }
46 }
47
48 int* prefix(string pattern, int size) {
49     int* lps = new int[size];
50     int len = 0, i = 1;
51     lps[0] = 0;
52     while (i < size) {
53         if (pattern[i] == pattern[len]) {
54             lps[i] = len;
55             len++;
```

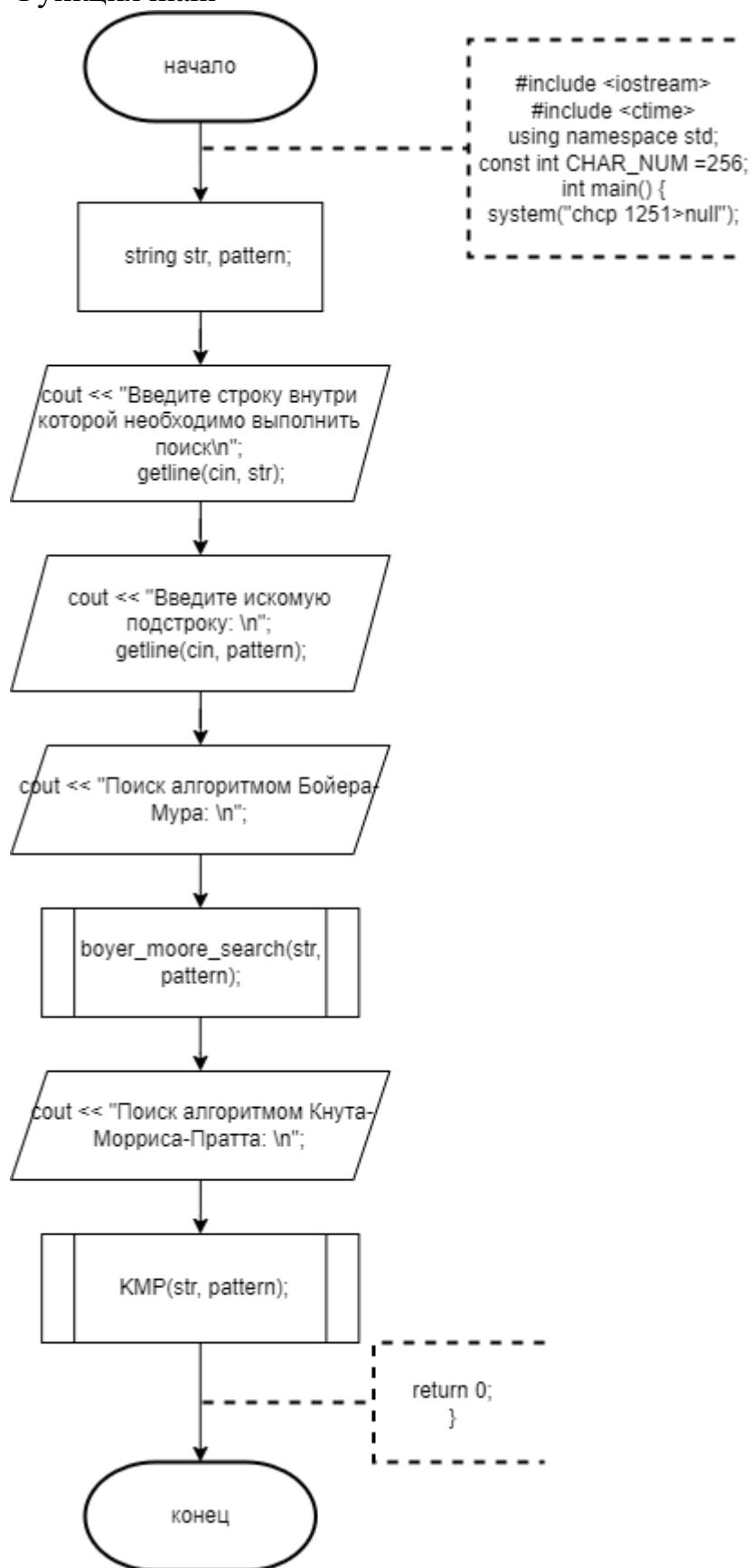
```

55         len++;
56         i++;
57     }
58     else {
59         if (len != 0) {
60             len = lps[len - 1];
61         }
62         else {
63             lps[i] = 0;
64             i++;
65         }
66     }
67 }
68 return lps;
69 }
70 void KMP(string str, string pattern) {
71     int str_size = str.size();
72     int pattern_size = pattern.size();
73     int* lps = prefix(pattern, pattern_size);
74     int pattern_index = 0;
75     int str_index = 0;
76
77     while ((str_size - str_index) >= (pattern_size - pattern_index)) {
78         if (pattern[pattern_index] == str[str_index]) {
79             str_index++;
80             pattern_index++;
81         }
82
83         if (pattern_index == pattern_size) {
84             cout << "Паттерн найден на позиции " << str_index - pattern_index + 1 << endl;
85             pattern_index = lps[pattern_index - 1];
86         }
87         else if (str_index < str_size and pattern[pattern_index] != str[str_index])
88         {
89             if (pattern_index != 0) {
90                 pattern_index = lps[pattern_index - 1];
91             }
92             else {
93                 str_index++;
94             }
95         }
96     }
97 }
98
99
100 int main() {
101     system("chcp 1251>null");
102
103     string str, pattern;
104
105     cout << "Введите строку внутри которой необходимо выполнить поиск\n";
106     getline(cin, str);
107
108     cout << "Введите искомую подстроку: \n";
109     getline(cin, pattern);
110
111     string str, pattern;
112
113     cout << "Введите строку внутри которой необходимо выполнить поиск\n";
114     getline(cin, str);
115
116     cout << "Введите искомую подстроку: \n";
117     getline(cin, pattern);
118
119     cout << "Поиск алгоритмом Бойера-Мура: \n";
120     boyer_moore_search(str, pattern);
121
122     cout << "Поиск алгоритмом Кнута-Морриса-Патта: \n";
123     KMP(str, pattern);
124
125     return 0;
126 }

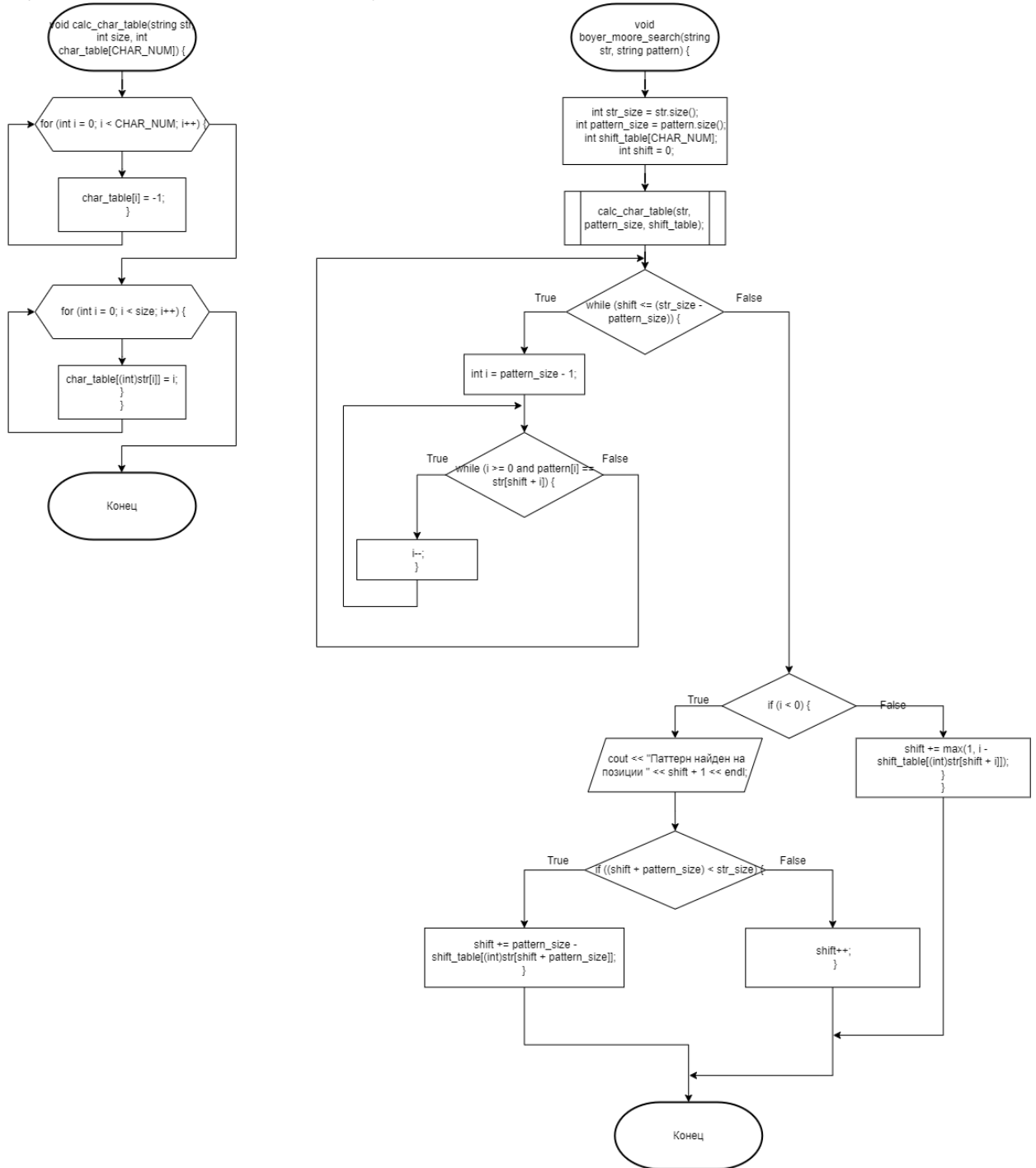
```

Блок схема

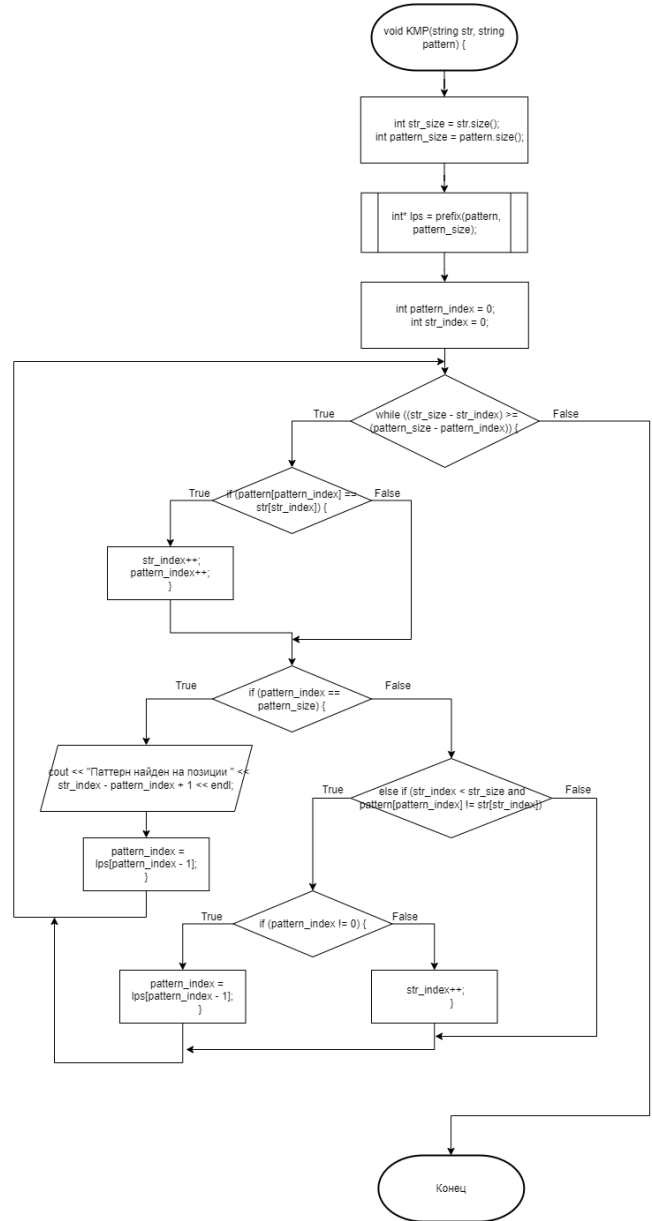
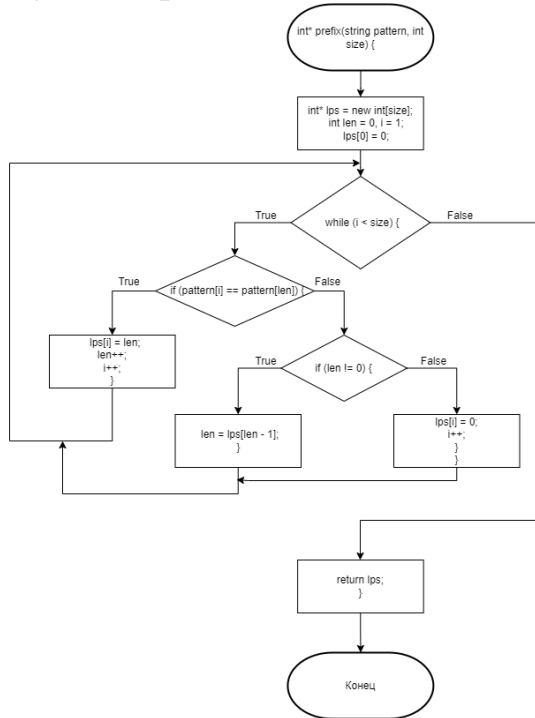
Функция main



Функции calc_char_table, boyer_moore_search



Функции prefix, KMP



Тесты

```
Введите строку внутри которой необходимо выполнить поиск
asldfasdbgk.asdkjbg.askdbg.kasdjghew;o
Введите искомую подстроку:
as
```

Поиск алгоритмом Бойера-Мура:

```
Паттерн найден на позиции 1
Паттерн найден на позиции 6
Паттерн найден на позиции 13
Паттерн найден на позиции 21
Паттерн найден на позиции 29
```

Поиск алгоритмом Кнута-Морриса-Пратта:

```
Паттерн найден на позиции 1
Паттерн найден на позиции 6
Паттерн найден на позиции 13
Паттерн найден на позиции 21
Паттерн найден на позиции 29
```

C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 2816) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

```
Введите строку внутри которой необходимо выполнить поиск
lldakbnjkeo/iwrobhelofaagsiughasidugbv,k
Введите искомую подстроку:
ob
```

Поиск алгоритмом Бойера-Мура:

```
Паттерн найден на позиции 16
```

Поиск алгоритмом Кнута-Морриса-Пратта:

```
Паттерн найден на позиции 16
```

C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 11280) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...