

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 17**

Выполнил студент гр. ИВТ-23-16
Давыдов Андрей Юрьевич

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка)

(подпись)

(дата)

г. Пермь, 2023

Постановка задачи

Поиски: линейный, интерполяционный, бинарный

Код программы:

```
#include <iostream>
#include <string>
using namespace std;

const int CHAR_NUM = 256;

void calc_char_table(const string& pattern, int char_table[]) {
    for (int i = 0; i < CHAR_NUM; ++i) {
        char_table[i] = -1;
    }
    for (size_t i = 0; i < pattern.size(); ++i) {
        char_table[static_cast<unsigned char>(pattern[i])] = i;
    }
}

void boyer_moore_search(const string& text, const string& pattern) {
    int char_table[CHAR_NUM];
    calc_char_table(pattern, char_table);
    int text_size = text.size();
    int pattern_size = pattern.size();
    int shift = 0;

    while (shift <= (text_size - pattern_size)) {
        int j = pattern_size - 1;
        while (j >= 0 && pattern[j] == text[shift + j]) {
            --j;
        }
        if (j < 0) {
            cout << "Паттерн найден на позиции: " << shift + 1 << endl;
            shift += (shift + pattern_size < text_size) ? pattern_size - char_table[static_cast<unsigned char>(text[shift + pattern_size])] : 1;
        }
        else {
            shift += max(1, j - char_table[static_cast<unsigned char>(text[shift + j])]);
        }
    }
}

void prefix(const string& pattern, int* lps) {
    int size = pattern.size();
    lps[0] = 0;
    int length = 0;
    int i = 1;

    while (i < size) {
        if (pattern[i] == pattern[length]) {
            ++length;
            lps[i] = length;
            ++i;
        }
        else {
            if (length != 0) {
                length = lps[length - 1];
            }
            else {
                lps[i] = 0;
                ++i;
            }
        }
    }
}

void KMP(const string& text, const string& pattern) {
    int str_size = text.size();
    int pattern_size = pattern.size();
    int* lps = new int[pattern_size];
    prefix(pattern, lps);
    int i = 0; // индекс для text
    int j = 0; // индекс для pattern

    while (i < str_size) {
        if (pattern[j] == text[i]) {
            ++j;
            ++i;
        }
        if (j == pattern_size) {
            cout << "Паттерн найден на позиции: " << i - j + 1 << endl;
            j = lps[j - 1];
        }
        else if (i < str_size && pattern[j] != text[i]) {
            if (j != 0) {
                j = lps[j - 1];
            }
            else {
                ++i;
            }
        }
    }

    delete[] lps;
}
```

```
int main() {
    system("chcp 1251>null");
    string text, pattern;
    cout << "Введите строку: ";
    getline(cin, text);
    cout << "Введите подстроку для поиска: ";
    getline(cin, pattern);

    cout << "Поиск алгоритмом Бойера-Мура:" << endl;
    boyer_moore_search(text, pattern);

    cout << "Поиск алгоритмом Кнута-Морриса-Пратта:" << endl;
    KMP(text, pattern);

    return 0;
}
```

Тест программы

```
Введите строку: всем привет
Введите подстроку для поиска: привет
Поиск алгоритмом Бойера-Мура:
Паттерн найден на позиции: 6
Поиск алгоритмом Кнута-Морриса-Пратта:
Паттерн найден на позиции: 6
```