

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт Космических и информационных технологий  
институт  
Кафедра «Информатика»  
кафедра

## ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Задание 1 - Поиск образа в строке  
тема

Преподаватель

\_\_\_\_\_  
подпись, дата

Р.Ю. Царев

инициалы, фамилия

Студент КИ18-17/16 031831229

номер группы, зачетной книжки

\_\_\_\_\_  
подпись, дата

В.А. Прекель

инициалы, фамилия

Красноярск 2020

## СОДЕРЖАНИЕ

Содержание .....	2
1 Цель работы с постановкой задачи .....	3
1.1 Цель работы .....	3
1.2 Задача работы .....	3
2 Описание реализованного алгоритма .....	3
3 Описание программы (листинги кода) .....	4
3.1 Lab_01/Lab_01_Console/main.cpp .....	4
3.2 Lab_01/Lab_01_Lib/BoyerMoore.h .....	5
3.3 Lab_01/Lab_01_Lib/BoyerMoore.cpp .....	5
3.4 Lab_01/Lab_01_LibTests/BoyerMooreTests.cpp .....	8
4 Результаты работы программы .....	9

## **1 Цель работы с постановкой задачи**

### **1.1 Цель работы**

Поиск образа в строке.

### **1.2 Задача работы**

Написать программу поиска образа в строке по методу Кнута, Морриса и Пратта либо Боуера и Мура (по выбору студента). Предусмотреть возможность существования в образе пробела. Ввести опцию чувствительности / нечувствительности к регистру.

Требования к выполнению лабораторной работы:

1. Строгое соответствие программы и результатов ее работы с полученным заданием.
2. Самостоятельное тестирование и отладка программы.
3. Устойчивость работы программы при любых воздействиях, задаваемых пользователем через интерфейс программы.
4. Предоставление демонстрационного примера и исходного текста программы для защиты.
5. Предоставление отчета по практическому заданию, содержащего описание реализованного алгоритма, программы, результатов работы программы (отчет необходимо загрузить на сайт курса).

## **2 Описание реализованного алгоритма**

Реализован алгоритм Боуера-Мура.

### 3 Описание программы (листинги кода)

#### 3.1 Lab\_01/Lab\_01\_Console/main.cpp

```
#include <cstdio>
#include <iostream>
#include <algorithm>

#ifdef _MSC_VER
#include <Windows.h>
#elif _WIN32
#include <windows.h>
#endif

#include "BoyerMoore.h"

int main(int argc, char** argv)
{
#ifdef _WIN32
    SetConsoleOutputCP(CP_UTF8);
    SetConsoleCP(CP_UTF8);
#endif

    std::cout << "Введите строку (в которой проводится поиск): ";
    std::string y;
    std::getline(std::cin, y);

    std::cout << "Введите образ (который ищется в строке): ";
    std::string x;
    std::getline(std::cin, x);

    std::cout << "Чувствительно к регистру? [Y/n]: ";
    std::string t;
    std::getline(std::cin, t);

    auto bm = BoyerMoore(y, x, t[0] == 'N' || t[0] == 'n');
    bm.Calculate();
    auto ans = bm.GetAnswer();

    std::cout << "Индексы образа в строке: ";
    for (auto i : *ans)
    {
        std::cout << i << ((i == ans->back()) ? "" : ", ");
    }
    std::cout << std::endl;

    std::cout << "Строка с помеченными образами: ";
    for (auto i = 0, j = -1; i < y.size(); i++)
    {
        if (std::count(ans->begin(), ans->end(), i))
        {
            std::cout << "[";
            j = x.size() - 1;
        }
        std::cout << y[i];
        if (j-- == 0)
            std::cout << "] ";
    }
    std::cout << std::endl;
}
```

```

        {
            std::cout << "];
        }
    }
    std::cout << std::endl;

    return 0;
}

```

### 3.2 Lab\_01/Lab\_01\_Lib/BoyerMoore.h

```

#ifndef BOYERMOORE_H
#define BOYERMOORE_H

#include <string>
#include <vector>

class BoyerMoore
{
private:
    std::string y; // строка
    std::string x; // образец

    std::vector<int>* borderArray;
    std::vector<int>* shiftArray;
    std::vector<int>* answer;

    void FullSuffixMatch();

    void PartialSuffixMatch();

    void BM();

public:
    BoyerMoore(std::string y, std::string x, bool insensitive);

    void Calculate();

    std::vector<int>* GetAnswer()
    {
        return answer;
    }

    ~BoyerMoore();
};

#endif //BOYERMOORE_H

```

### 3.3 Lab\_01/Lab\_01\_Lib/BoyerMoore.cpp

```

#include <algorithm>

#include "BoyerMoore.h"

```

```

void BoyerMoore::FullSuffixMatch()
{
    int n = x.size();
    int i = n;
    int j = n + 1;
    borderArray->at(i) = j;

    while (i > 0)
    {
        while (j <= n && x[i - 1] != x[j - 1])
        {
            if (shiftArray->at(j) == 0)
            {
                shiftArray->at(j) = j - i;
            }
            j = borderArray->at(j);
        }
        i--;
        j--;
        borderArray->at(i) = j;
    }
}

void BoyerMoore::PartialSuffixMatch()
{
    int n = x.size();
    int j;
    j = borderArray->at(0);

    for (int i = 0; i < n; i++)
    {
        if (shiftArray->at(i) == 0)
        {
            shiftArray->at(i) = j;
        }
        if (i == j)
        {
            j = borderArray->at(j);
        }
    }
}

void BoyerMoore::BM()
{
    int m = x.length();
    int n = y.length();

    if (m == 0)
    {
        answer->push_back(-1);
        return;
    }
    if (m == 1)
    {
        for (auto i = 0; i < n; i++)
        {
            if (y[i] == x[0])
            {

```

```

        answer->push_back(i);
    }
}
return;
}

borderArray->assign(m + 1, 0);
shiftArray->assign(m + 1, 0);

FullSuffixMatch();
PartialSuffixMatch();

int shift = 0;

while (shift <= (n - m))
{
    int j = m - 1;
    while (j >= 0 && x[j] == y[shift + j])
    {
        j--;
    }

    if (j < 0)
    {
        answer->push_back(shift);
        shift += shiftArray->at(0);
    }
    else
    {
        shift += shiftArray->at(j + 1);
    }
}

if (answer->empty())
{
    answer->push_back(-1);
}
}

BoyerMoore::BoyerMoore(std::string y, std::string x, bool insensitive)
{
    this->y = std::move(y);
    this->x = std::move(x);

    if (insensitive)
    {
        std::transform(this->y.begin(), this->y.end(), this->y.begin(),
            [](auto c)
            { return std::tolower(c); });
        std::transform(this->x.begin(), this->x.end(), this->x.begin(),
            [](auto c)
            { return std::tolower(c); });
    }

    borderArray = new std::vector<int>();
    shiftArray = new std::vector<int>();
    answer = new std::vector<int>();
}

void BoyerMoore::Calculate()

```

```

{
    BM();
}

BoyerMoore::~BoyerMoore()
{
    delete borderArray;
    delete shiftArray;
    delete answer;
}

```

### 3.4 Lab\_01/Lab\_01\_LibTests/BoyerMooreTests.cpp

```

#include <vector>

#include <gtest/gtest.h>

#include "BoyerMoore.h"

using namespace testing;

TEST(BoyerMooreTests, BoyerMooreTest1)
{
    auto bm = new BoyerMoore("rgsgfdsq125rerdsqs", "r", false);
    bm->Calculate();
    auto answer = bm->GetAnswer();

    ASSERT_EQ(answer->size(), 3);
    EXPECT_EQ(answer->at(0), 0);
    EXPECT_EQ(answer->at(1), 11);
    EXPECT_EQ(answer->at(2), 13);

    delete bm;
}

TEST(BoyerMooreTests, BoyerMooreTest2)
{
    auto bm = new BoyerMoore("123123", "23", false);
    bm->Calculate();
    auto answer = bm->GetAnswer();

    ASSERT_EQ(answer->size(), 2);
    EXPECT_EQ(answer->at(0), 1);
    EXPECT_EQ(answer->at(1), 4);

    delete bm;
}

TEST(BoyerMooreTests, BoyerMooreTest3)
{
    auto bm = new BoyerMoore("ABAAABCDDBBABCDEBCABC", "ABC", false);
    bm->Calculate();
    auto answer = bm->GetAnswer();

    ASSERT_EQ(answer->size(), 3);
    EXPECT_EQ(answer->at(0), 4);
    EXPECT_EQ(answer->at(1), 10);
}

```



```

    EXPECT_EQ(answer->at(2), 18);

    delete bm;
}

TEST(BoyerMooreTests, BoyerMooreTest4)
{
    auto bm = new BoyerMoore("GCATCGCAGAGAGTATACAGTACG", "GCAGAGAG", false);
    bm->Calculate();
    auto answer = bm->GetAnswer();

    ASSERT_EQ(answer->size(), 1);
    EXPECT_EQ(answer->at(0), 5);

    delete bm;
}

TEST(BoyerMooreTests, BoyerMooreTest5)
{
    auto bm = BoyerMoore("GCATCGCAGAGAGTATACAGTACG", "GcagAGAG", true);
    bm.Calculate();
    auto answer = bm.GetAnswer();

    ASSERT_EQ(answer->size(), 1);
    EXPECT_EQ(answer->at(0), 5);
}

```

## 4 Результаты работы программы

```

vladislav@DESKTOP-ODR692H: /mnt/c/Users/vladislav/Projects/AlgorithmsAndDataStructures/cmake-build-debug/Lab_01/Lab_01_Console
nsole$ ./Lab_01_Console
Введите строку (в которой проводится поиск): qwewqewfewdgweswwe
Введите образ (который ищется в строке): we
Чувствительно к регистру? [Y/n]:
Индексы образа в строке: 1, 12, 16
Строка с помеченными образами: q[we]wqewfewdg[we]sw[we]
vladislav@DESKTOP-ODR692H: /mnt/c/Users/vladislav/Projects/AlgorithmsAndDataStructures/cmake-build-debug/Lab_01/Lab_01_Console
nsole$ ./Lab_01_Console
Введите строку (в которой проводится поиск): qdsaFFDd dsDDDDsf Dff dfdfkljsjd dJSHDFjdsjdhjHH dsd213 s
Введите образ (который ищется в строке): D
Чувствительно к регистру? [Y/n]: n
Индексы образа в строке: 8, 17, 21, 33, 49
Строка с помеченными образами: qdsaFFDd[ d]sDDDDsf[ D]Fff[ d]fdfkljsjd [ d]JSHDFjdsjdhjHH[ d]sd213 s
vladislav@DESKTOP-ODR692H: /mnt/c/Users/vladislav/Projects/AlgorithmsAndDataStructures/cmake-build-debug/Lab_01/Lab_01_Console
nsole$ ./Lab_01_Console
Введите строку (в которой проводится поиск): qweeeqwwweEeeEEeewegfdfeE
Введите образ (который ищется в строке): e
Чувствительно к регистру? [Y/n]: n
Индексы образа в строке: 2, 3, 4, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 24, 25
Строка с помеченными образами: qw[e][e][e]qwww[e][E][e][e][E][E][e][e]w[e][e]gfdfe[E]
vladislav@DESKTOP-ODR692H: /mnt/c/Users/vladislav/Projects/AlgorithmsAndDataStructures/cmake-build-debug/Lab_01/Lab_01_Console
nsole$ ./Lab_01_Console
Введите строку (в которой проводится поиск): 123123234532456345214654768456585
Введите образ (который ищется в строке): 21
Чувствительно к регистру? [Y/n]:
Индексы образа в строке: 18
Строка с помеченными образами: 123123234532456345[21]4654768456585
vladislav@DESKTOP-ODR692H: /mnt/c/Users/vladislav/Projects/AlgorithmsAndDataStructures/cmake-build-debug/Lab_01/Lab_01_Console
nsole$

```

Рисунок 1 – Запуск 1 (WSL Ubuntu 18.04, gcc)

```
PowerShell 7 (x64)
PS C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\cmake-build-debug-mingw\Lab_01\Lab_01_Console> .\Lab_01_Console.exe
Введите строку (в которой проводится поиск): qwewqetretwebhtrrweweerwegfhhwerwe
Введите образ (который ищется в строке): we
Чувствительно к регистру? [Y/n]:
Индексы образа в строке: 1, 10, 18, 21, 25, 31, 34
Строка с помеченными образами: q[we]wqetret[we]bhtrrw[we]e[we]er[we]gfhh[we]r[we]
PS C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\cmake-build-debug-mingw\Lab_01\Lab_01_Console> .\Lab_01_Console.exe
Введите строку (в которой проводится поиск): fsfwewewefsdwd FSDAFFDFdfdfDFEG dfdDD dsfs DDdDFDFFDFdFDfd
Введите образ (который ищется в строке): fd
Чувствительно к регистру? [Y/n]: n
Индексы образа в строке: 22, 24, 26, 28, 30, 36, 38, 53, 57, 59, 61, 63, 65
Строка с помеченными образами: fsfwewewefsdwd FSDAF[FD][Fd][Fd][Fd][FD]FEG [fd][fd]DD dsfs DDdDF[FD]FF[FD][fd][FD][Fd][fd]
PS C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\cmake-build-debug-mingw\Lab_01\Lab_01_Console> .\Lab_01_Console.exe
Введите строку (в которой проводится поиск): QasddfsffsdSDsSSdsds
Введите образ (который ищется в строке): D
Чувствительно к регистру? [Y/n]: n
Индексы образа в строке: 3, 4, 6, 11, 12, 14, 18, 19, 21
Строка с помеченными образами: Qas[d][d]f[d]sffs[d][D]S[D]sSS[d][d]s[d]s
PS C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\cmake-build-debug-mingw\Lab_01\Lab_01_Console> .\Lab_01_Console.exe
Введите строку (в которой проводится поиск): asdfsd
Введите образ (который ищется в строке): aaasdsadsfsd
Чувствительно к регистру? [Y/n]:
Индексы образа в строке: -1
Строка с помеченными образами: asdfsd
PS C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\cmake-build-debug-mingw\Lab_01\Lab_01_Console>
```

Рисунок 2 – Запуск 2 (Windows 10, mingw)

```
PowerShell 7 (x64)
PS C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\cmake-build-debug-mingw\Lab_01\Lab_01_LibTests> .\Lab_01_LibTests.exe
Running main() from C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\googletest\googletest\src\gtest_main.cc
=====
Running 5 tests from 1 test suite.
Global test environment set-up.
5 tests from BoyerMooreTests
RUN      OK      BoyerMooreTests.BoyerMooreTest1 (0 ms)
RUN      OK      BoyerMooreTests.BoyerMooreTest2 (0 ms)
RUN      OK      BoyerMooreTests.BoyerMooreTest3 (0 ms)
RUN      OK      BoyerMooreTests.BoyerMooreTest4 (0 ms)
RUN      OK      BoyerMooreTests.BoyerMooreTest5 (0 ms)
=====
5 tests from BoyerMooreTests (24 ms total)

Global test environment tear-down
=====
5 tests from 1 test suite ran. (37 ms total)
PASSED   5 tests.
PS C:\Users\vladislav\Projects\AlgorithmsAndDataStructures\cmake-build-debug-mingw\Lab_01\Lab_01_LibTests>
```

Рисунок 3 – Запуск тестов