Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе «Поиск Бойера-Мура»**

Дисциплина: Основы алгоритмизации и программирования

Выполнил:

Студентка группы ИВТ-22-2б

Мифтахов Марат Ринатович

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2023**

**Постановка задачи**

Реализовать программу, которая будет искать подстроку в строку и выводить соответствующее сообщение.

**Анализ задачи**

1. Первоначально строится таблица смещений для искомой подстроки.

int needle\_table[256];

for (i = 0; i < 256; i++)

needle\_table[i] = needle\_len;

for (i = 1; i < needle\_len; i++)

needle\_table[needle[i-1]] = needle\_len-i;

1. Далее идет совмещение начала строки и подстроки и начинается проверка с последнего символа подстроки.

Если последний символ подстроки и соответствующий ему при наложении символ строки не совпадают, подстрока сдвигается относительно строки на величину, полученную из таблицы смещений, и [снова проводится сравнение](http://www.topuch.ru/poka-ya-pomnyu--ya-jivu/index.html), начиная с последнего символа подстроки.

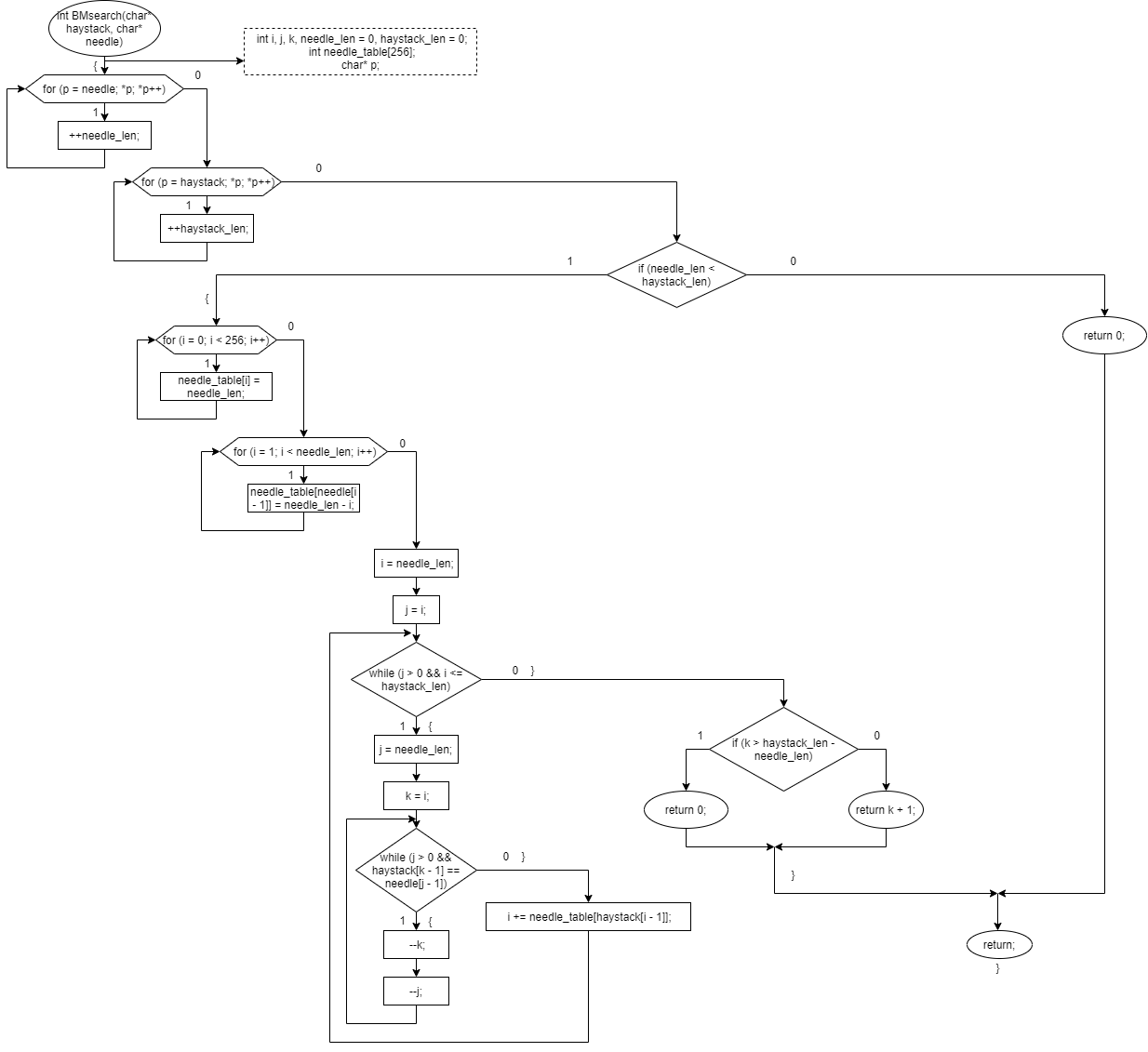
Если же символы совпадают, производится сравнение предпоследнего символа подстроки и т.д.

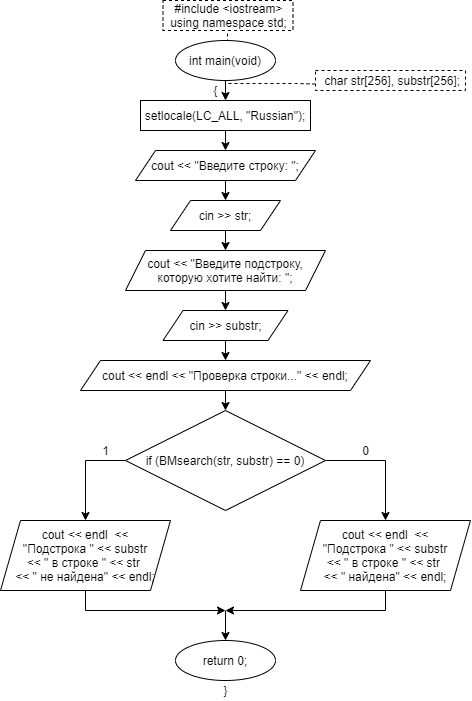
Если все символы подстроки совпали с наложенными символами строки, значит, найдена подстрока и поиск окончен.

Если же какой-то (не последний) символ подстроки не совпадает с соответствующим символом строки, далее производим сдвиг подстроки на один символ вправо и снова начинаем проверку с последнего символа.

1. В главной функции у пользователя запрашивается строка и подстрока, которую нужно найти в строке.
2. А затем вызывается функция поиска подстроки в строке по алгоритму Бойера-Мура *BMsearch(str, substr)*, в качестве параметров передаются строка и подстрока.
3. Если подстрока была найдена, то выводится сообщение о том, данная подстрока найдена в заданной строке, если же подстрока не была найдена, то так же выводится сообщение о том, что она не была найдена.

**Блок-схема**





**Исходный код**

#include <iostream>

using namespace std;

int BMsearch(char\* haystack, char\* needle)

{

int i, j, k, needle\_len = 0, haystack\_len = 0;

int needle\_table[256];

char\* p;

for (p = needle; \*p; \*p++)

++needle\_len;

for (p = haystack; \*p; \*p++)

++haystack\_len;

if (needle\_len < haystack\_len)

{

for (i = 0; i < 256; i++)

needle\_table[i] = needle\_len;

for (i = 1; i < needle\_len; i++)

needle\_table[needle[i - 1]] = needle\_len - i;

i = needle\_len;

j = i;

while (j > 0 && i <= haystack\_len)

{

j = needle\_len;

k = i;

while (j > 0 && haystack[k - 1] == needle[j - 1])

{

--k;

--j;

}

i += needle\_table[haystack[i - 1]];

}

if (k > haystack\_len - needle\_len)

return 0;

else return k + 1;

}

else return 0;

}

int main(void)

{

setlocale(LC\_ALL, "Russian");

char str[256], substr[256];

cout << "Введите строку: ";

cin >> str;

cout << "Введите подстроку которую хотите найти: ";

cin >> substr;

cout << endl;

cout << "Проверка строки..." << endl;

if (BMsearch(str, substr) == 0)

{

cout << endl << "Подстрока " << substr << " в строке " << str << " не найдена" << endl;

}

else

{

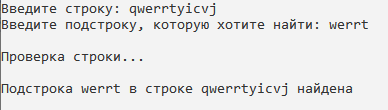
cout << endl << "Подстрока " << substr << " в строке " << str << " найдена" << endl;

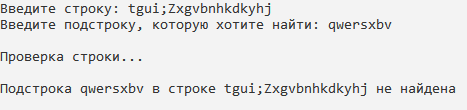
}

return 0;

}

**Скриншоты**





**Анализ (объяснение скриншотов)**

Программа запрашивает у пользователя строку и подстроку, которую нужно найти. Затем происходит проверка строки, и программа выводит соответствующее сообщение о том, найдена ли подстрока в заданной строке.