МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа:**

"Поиск методом БМ"

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Дана строка и подстрока, которую нужно найти. Выполнить задание используя метод БМ.

**Анализ задачи:**

Алгоритм поиска строки Бойера-Мура – это алгоритм общего назначения, предназначенный для поиска подстроки в строке.

Преимущество этого алгоритма в том, что ценой некоторого количества предварительных вычислений над шаблоном (но не над строкой, в которой ведётся поиск), шаблон сравнивается с исходным текстом не во всех позициях — часть проверок пропускается как заведомо не дающая результата.

Совмещается начало текста (строки) и шаблона, проверка начинается с последнего символа шаблона. Если символы совпадают, производится сравнение предпоследнего символа шаблона и т. д. Если все символы шаблона совпали с наложенными символами строки, значит, подстрока найдена, и выполняется поиск следующего вхождения подстроки. Если же какой-то символ шаблона не совпадает с соответствующим символом строки, шаблон сдвигается на несколько символов вправо, и проверка снова начинается с последнего символа.

**Код на С++ с комментариями:**

#include<iostream>

#include<string>

using namespace std;

int BMsearch(string str, string substr) {

int strl, substrl, res = -1;

strl = str.size(); substrl = substr.size();

if (strl != 0 && substrl != 0) {

int i, Pos;

int bias[256];// массив смещения

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

bias[i] = substrl;//присвоить каждому элементу массива смещения длину подстроки

}

for (i = substrl - 2; i >= 0; i--) { // Корректируем массив смещения

if (bias[int((unsigned char)substr[i])] == substrl) {

bias[int((unsigned char)substr[i])] = substrl - i - 1;

}

}

Pos = substrl - 1; //позиция последнего символа подстроки относительно строки

while (Pos < strl) { //выполняется тело цикла, пока есть подстрока в строке

if (substr[substrl - 1] != str[Pos]) {//последний символ подстроки и символ стороки

Pos += bias[int((unsigned char)str[Pos])];//сдвиг по таблице

}

else {

for (i = substrl - 1; i >= 0; i--) { // проход по символам подстроки начиная с предпоследнего

if (substr[i] == str[Pos - substrl + 1 + i]) {// сравнение символа подстроки и символа стороки

if (i == 0) { // если прошли всю подстроку

return Pos - substrl + 1; // считаем индекс

}

}

else {

Pos += bias[int((unsigned char)str[Pos - substrl + 1 + i])];

break;

}

}

}

}

}

return res;

}

int main()

{

setlocale(LC\_ALL, "ru");

string str, templ;

cout << "Введите строку: ";

cin >> str;

cout << "Введите подстроку: ";

cin >> templ;

int index = BMsearch(str, templ);

if (index != -1)

{

cout << "Подстрока под индексом: " << index << endl;

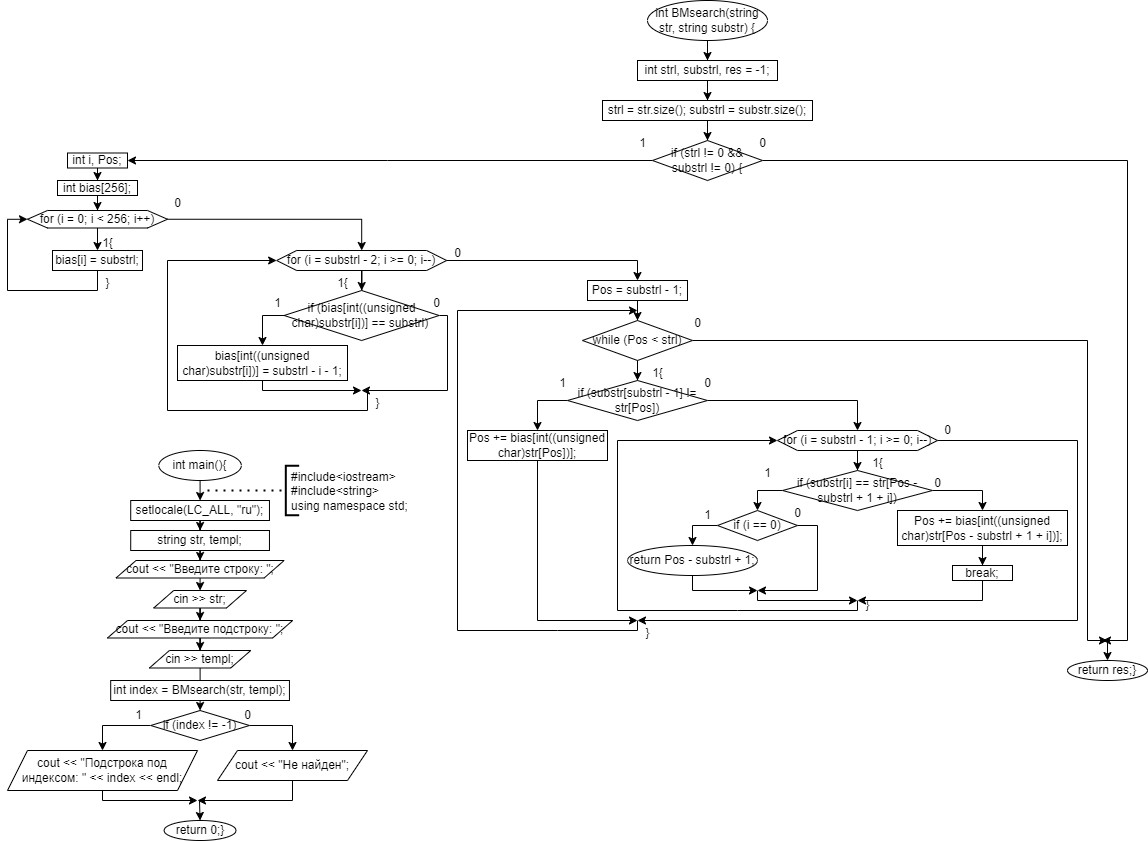
}

else

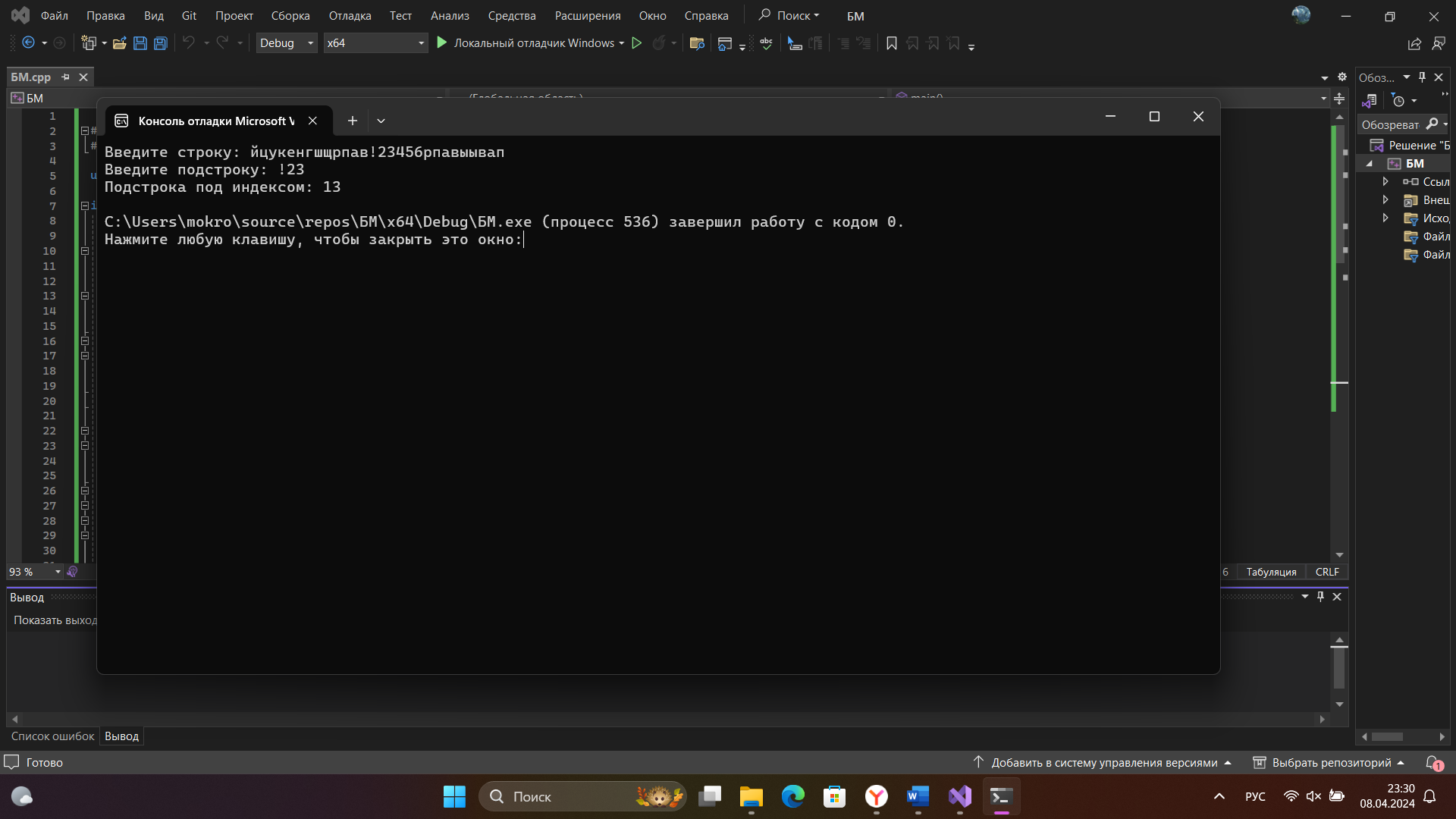
cout << "Не найден";

return 0;

}

**Блок схема:**

**Скриншот работы программы:**



**Вывод:**

Программа работает успешно.

**Скриншот с GitHab:**