Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил

студент группы ИВТ 21-1Б

Лебедев Т.А.

Проверил

Яруллин Д.В.

Пермь, 2022

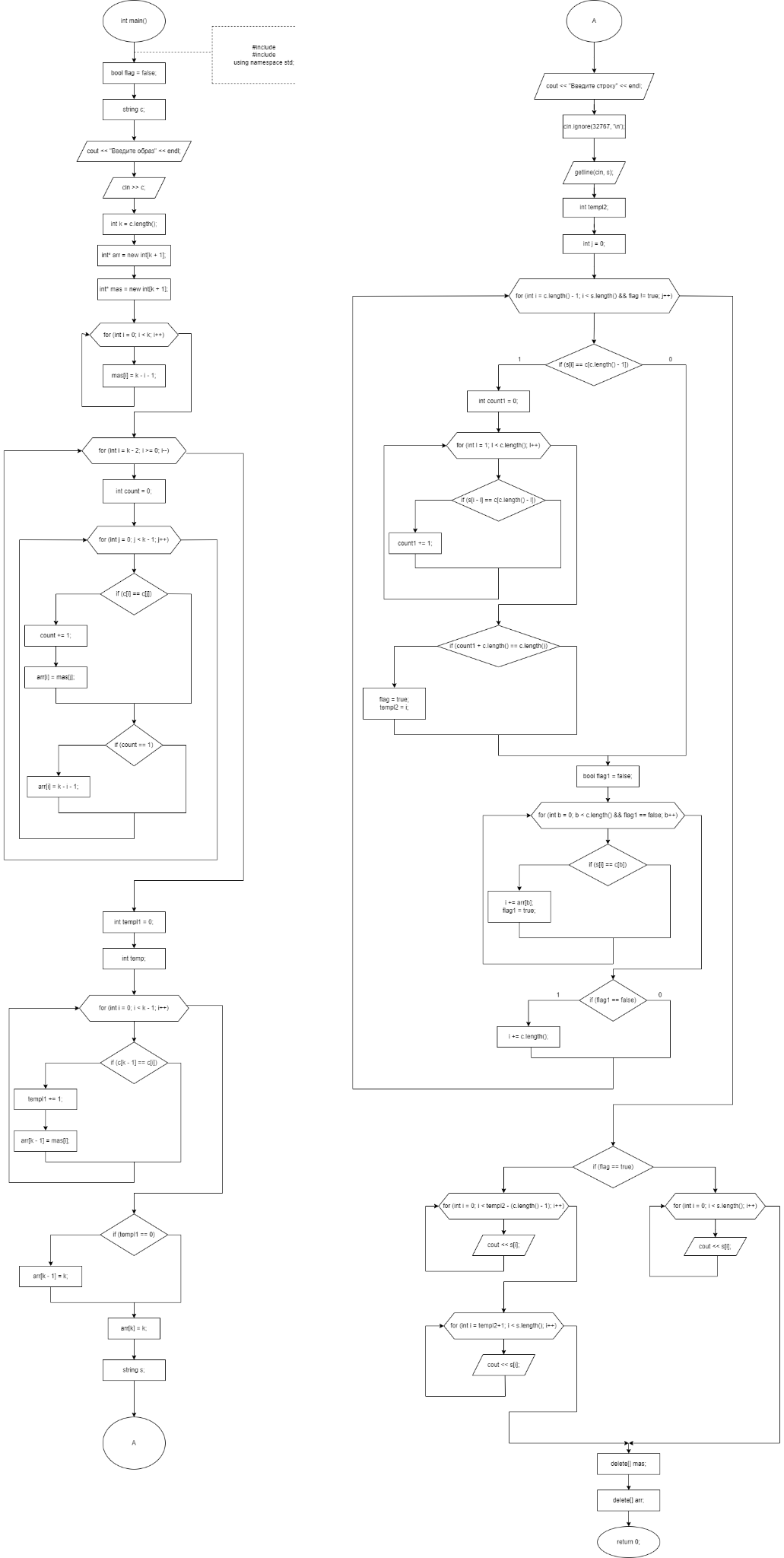
**Постановка задачи:**

Дана строка и подстрока, которую нужно найти. Выполнить задание в соответствии со своим вариантом, используя указанный метод поиска. Значения строки и подстроки любые (если иного не указано по заданию). Найти и удалить первое вхождение подстроки strl

**Анализ задачи:**

1. Алгоритм поиска строки Бойера — Мура — алгоритм общего назначения, предназначенный для поиска подстроки в строке. 2. 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. 3. 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.

**Блок-схема:**

****

**Код программы:**

#include<iostream>

#include<string>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

bool flag = false;

string c;

cout << "Введите образ" << endl;

cin >> c;

int k = c.length();

int\* arr = new int[k + 1];

int\* mas = new int[k + 1];

for (int i = 0; i < k; i++)

{

mas[i] = k - i - 1;

}

for (int i = k - 2; i >= 0; i--)

{

int count = 0;

for (int j = 0; j < k - 1; j++)

{

if (c[i] == c[j])

{

count += 1;

arr[i] = mas[j];

}

}

if (count == 1)

{

arr[i] = k - i - 1;

}

}

int templ1 = 0;

int temp;

for (int i = 0; i < k - 1; i++)

{

if (c[k - 1] == c[i])

{

templ1 += 1;

arr[k - 1] = mas[i];

}

}

if (templ1 == 0)

{

arr[k - 1] = k;

}

arr[k] = k;

string s;

cout << "Введите строку" << endl;

cin.ignore(32767, '\n');

getline(cin, s);

int templ2;

int j = 0;

for (int i = c.length() - 1; i < s.length() && flag != true; j++)

{

if (s[i] == c[c.length() - 1])

{

int count1 = 0;

for (int l = 1; l < c.length(); l++)

{

if (s[i - l] == c[c.length() - l])

{

count1 += 1;

}

}

if (count1 + c.length() == c.length())

{

flag = true;

templ2 = i;

break;

}

}

bool flag1 = false;

for (int b = 0; b < c.length() && flag1 == false; b++)

{

if (s[i] == c[b])

{

i += arr[b];

flag1 = true;

}

}

if (flag1 == false)

{

i += c.length();

}

}

if (flag == true)

{

for (int i = 0; i < templ2 - (c.length() - 1); i++)

{

cout << s[i];

}

for (int i = templ2+1; i < s.length(); i++)

{

cout << s[i];

}

}

else

{

for (int i = 0; i < s.length(); i++)

{

cout << s[i];

}

}

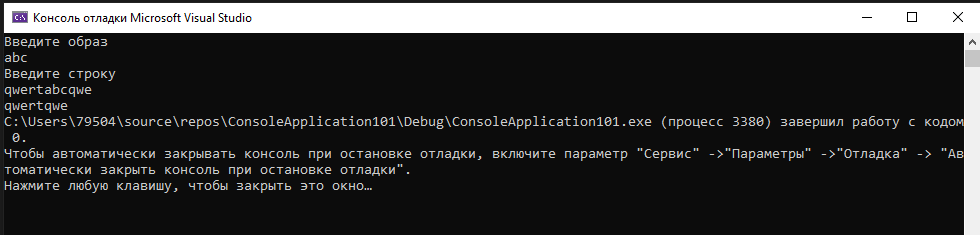
delete mas;

delete arr;

return 0;

}

**Скриншоты результатов:**

****