МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа:**

"Поиск методом КМП"

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 КМП.

**Анализ задачи:**

Поиск Кнута-Морриса-Пратта – это эффективный алгоритм, осуществляющий поиск подстроки в строке. Время работы алгоритма линейно зависит от объёма входных данных, то есть разработать асимптотически более эффективный алгоритм невозможно.

**Код на С++:**

#include<iostream>

#include<string>

using namespace std;

int KMPSearch(string& str, string& templ)

{

int strlen, templlen;

strlen = str.size();

templlen = templ.size();

if (strlen == 0)

cout << "Неверно задана строка\n";

else if (templlen == 0)

cout << "Неверно задана подстока\n";

else

{

int i = 0, j = -1;

int\* d = new int[100];

d[0] = -1;

while (i < templlen - 1)

{

while (j >= 0 && templ[i] != templ[j])

j = d[j];

i++;

j++;

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

d[i] = d[j];

else

d[i] = j;

}

i = 0, j = 0;

while ((j < templlen) && (i < strlen))

{

while (j >= 0 && str[i] != templ[j])

j = d[j];

i++;

j++;

}

delete[] d;

if (j == templlen)

return i - j;

else

return -1;

}

}

int main()

{

setlocale(LC\_ALL, "ru");

string str, templ;

cout << "Введите строку: ";

cin >> str;

cout << "Введите подстроку: ";

cin >> templ;

int index = KMPSearch(str, templ);

if (index != 1)

{

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

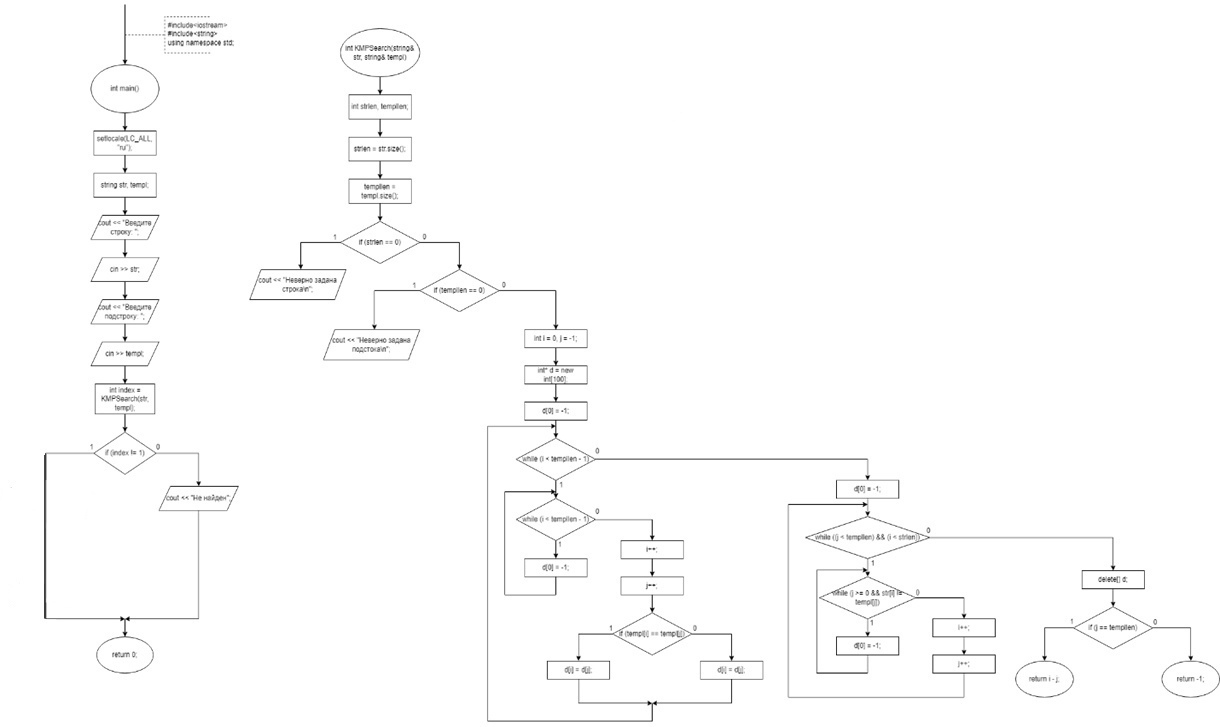
}

else

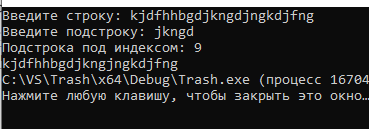
cout << "Не найден";

return 0;

}

**Блок схема:**

**Скриншот работы программы:**



**Вывод:**

Программа работает успешно

**Скриншот с GitHab:**