Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**Отчет по Лабораторной работе метод Кнута – Морриса – Пратта**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

Выполнил студент гр. ИВТ-21-1б

Ахунов Руслан Булатович

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Денис Владимирович

(оценка) (подпись)

г. Пермь - 2022

**Постановка задачи:**

Для метода Бойера – Мура и Кнута – Морриса – Пратта: 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).

Вариант 2

Найти подстроку strl, переместить её в начало строки.

**Анализ задачи:**

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

        {

            j = d[j];

        }

        i++;

        j++;

        if (sbstr[i] == sbstr[j])

        {

            d[i] = d[j];

        }

        else

        {

            d[i] = j;

        }

        i = 0;

        j = 0;

Подготовка массива смещений, идет проверка префикс-функций.

for (i = 0, j = 0; (i <= st - 1) && (j <= sbst - 1); i++, j++)

        {

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

            {

                j = d[j];

            }

        }

        delete[] d;

        if (j == sbst)

        {

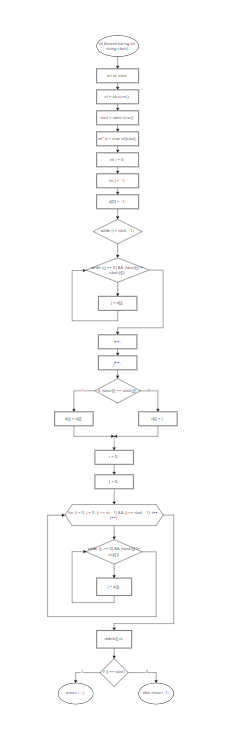
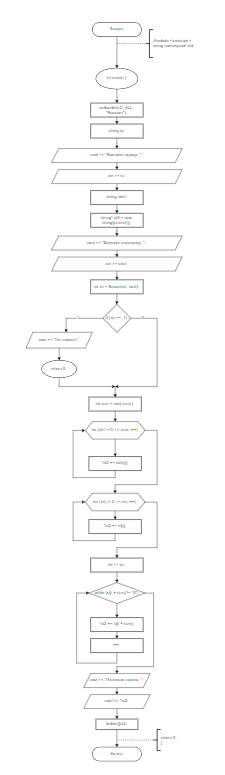
            return i - j;

        }

        else return -1;

Поиск подстроки в строке и возвращение индекса найденного символа.

**Блок-схема**

****

**Код программы**

#include <iostream>

#include <fstream>

#include <iostream>

#include <string>

using namespace std;

int Search(string str, string sbstr)

{

    int st, sbst;

    st = str.size();

    sbst = sbstr.size();

    int\* d = new int[sbst];

    int i = 0;

    int j = -1;

    d[0] = -1;

    while (i < sbst - 1)

    {

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

        {

            j = d[j];

        }

        i++;

        j++;

        if (sbstr[i] == sbstr[j])

        {

            d[i] = d[j];

        }

        else

        {

            d[i] = j;

        }

        i = 0;

        j = 0;

        for (i = 0, j = 0; (i <= st - 1) && (j <= sbst - 1); i++, j++)

        {

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

            {

                j = d[j];

            }

        }

        delete[] d;

        if (j == sbst)

        {

            return i - j;

        }

        else return -1;

    }

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    string st;

    cout << "Введите строку: ";

    cin >> st;

    string sbst;

    string\* st2 = new string[st.size()];

    cout << "Введите подстроку: ";

    cin >> sbst;

    int ist = Search(st, sbst);

    if (ist == -1)

    {

        cout << "Не найден";

        return 0;

    }

    int size = sbst.size();

    for (int i = 0; i < size; i++)

    {

        \*st2 += sbst[i];

    }

    for (int i = 0; i < ist; i++)

    {

        \*st2 += st[i];

    }

    int i = ist;

    while (st[i + size] != '\0')

    {

        \*st2 += st[i + size];

        i++;

    }

    cout << "Итоговая строка: ";

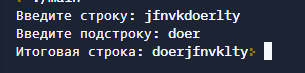
    cout << \*st2;

    delete[]st2;

    return 0;

}

**Вывод программы**

****