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 Бойера – Мура**

**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:**

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 и удалить первое вхождение подстроки strl

**Анализ задачи:**

int pos;

        int b[256];

        for (int i = 0; i < 256; i++)

        {

            b[i] = sbst;

        }

Подготавливаем таблицу смещения, заранее ставя в нее максимально возможное смещение, он равен длине подстроки.

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

        {

            if (b[int((unsigned char)sbstr[i])] == sbst)

            {

                b[int((unsigned char)sbstr[i])] = sbst - i - 1;

            }

        }

Корректируем массив смещения, каждой букву, благодаря ASCII мы присваиваем определенное смещение. I = sbst – 2, так как цикл идет до 0, и самый последний символ мы не рассматриваем.

pos = sbst - 1;

        while (pos < st)

        {

            if (sbstr[sbst - 1] != str[pos])

            {

                pos += b[int((unsigned char)str[pos])];

            }

            else

            {

                for (int i = sbst - 1; i >= 0; i--)

                {

                    if (sbstr[i] != str[pos - sbst + i + 1])

                    {

                        pos += b[int((unsigned char)str[pos - sbst + i + 1])];

                        break;

                    }

                    else if (i == 0)

                    {

                        return pos - sbst + 1;

                    }

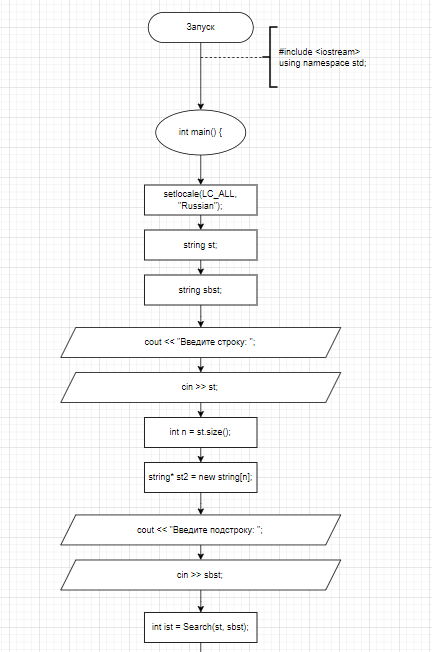
                }

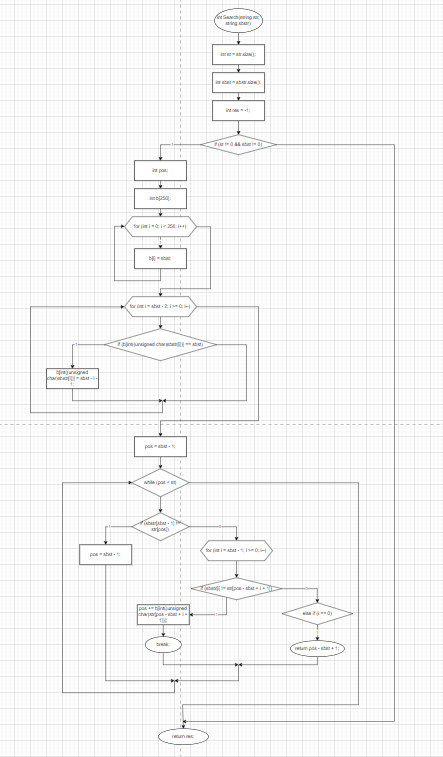
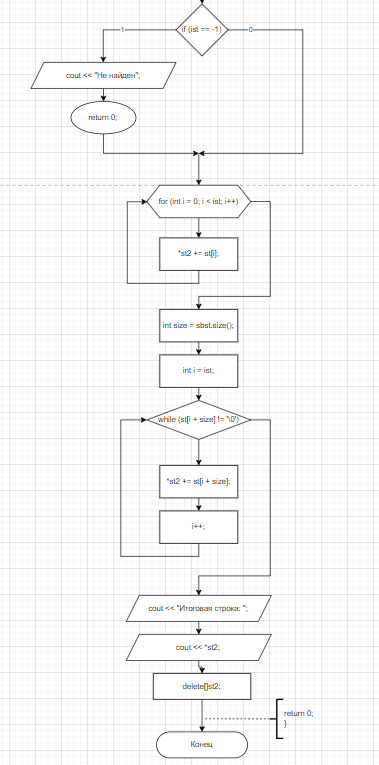
            }

        }

Начинается проход по всей строке.

**Блок-схема**

****

****

**Код программы**

#include <iostream>

#include <string>

using namespace std;

int Search(string str, string sbstr)

{

    int st = str.size();

    int sbst = sbstr.size();

    int res = -1;

    if (st != 0 && sbst != 0)

    {

        int pos;

        int b[256];

        for (int i = 0; i < 256; i++)

        {

            b[i] = sbst;

        }

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

        {

            if (b[int((unsigned char)sbstr[i])] == sbst)

            {

                b[int((unsigned char)sbstr[i])] = sbst - i - 1;

            }

        }

        pos = sbst - 1;

        while (pos < st)

        {

            if (sbstr[sbst - 1] != str[pos])

            {

                pos += b[int((unsigned char)str[pos])];

            }

            else

            {

                for (int i = sbst - 1; i >= 0; i--)

                {

                    if (sbstr[i] != str[pos - sbst + i + 1])

                    {

                        pos += b[int((unsigned char)str[pos - sbst + i + 1])];

                        break;

                    }

                    else if (i == 0)

                    {

                        return pos - sbst + 1;

                    }

                }

            }

        }

    }

    return res;

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    string st;

    string sbst;

    cout << "Введите строку: ";

    cin >> st;

    int n = st.size();

    string\* st2 = new string[n];

    cout << "Введите подстроку: ";

    cin >> sbst;

    int ist = Search(st, sbst);

    if (ist == -1)

    {

        cout << "Не найден";

        return 0;

    }

    for (int i = 0; i < ist; i++)

    {

        \*st2 += st[i];

    }

    int size = sbst.size();

    int i = ist;

    while (st[i + size] != '\0')

    {

        \*st2 += st[i + size];

        i++;

    }

    cout << "Итоговая строка: ";

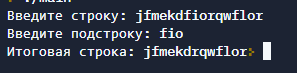
    cout << \*st2;

    delete[]st2;

    return 0;

}

**Вывод программы**

****