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»

**О Т Ч Е Т**

**по лабораторной работе №14 (Кнута-Морриса-Пратта)**

**по дисциплине**

**«Основы алгоритмизации и программирования» семестр 2**

Выполнил студент гр. ИВТ-21-1б

Ипатов Дмитрий Сергеевич

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Д.В.

(оценка) (подпись)

г. Пермь-2022

**Постановка задачи:**

Для методаКнута-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).

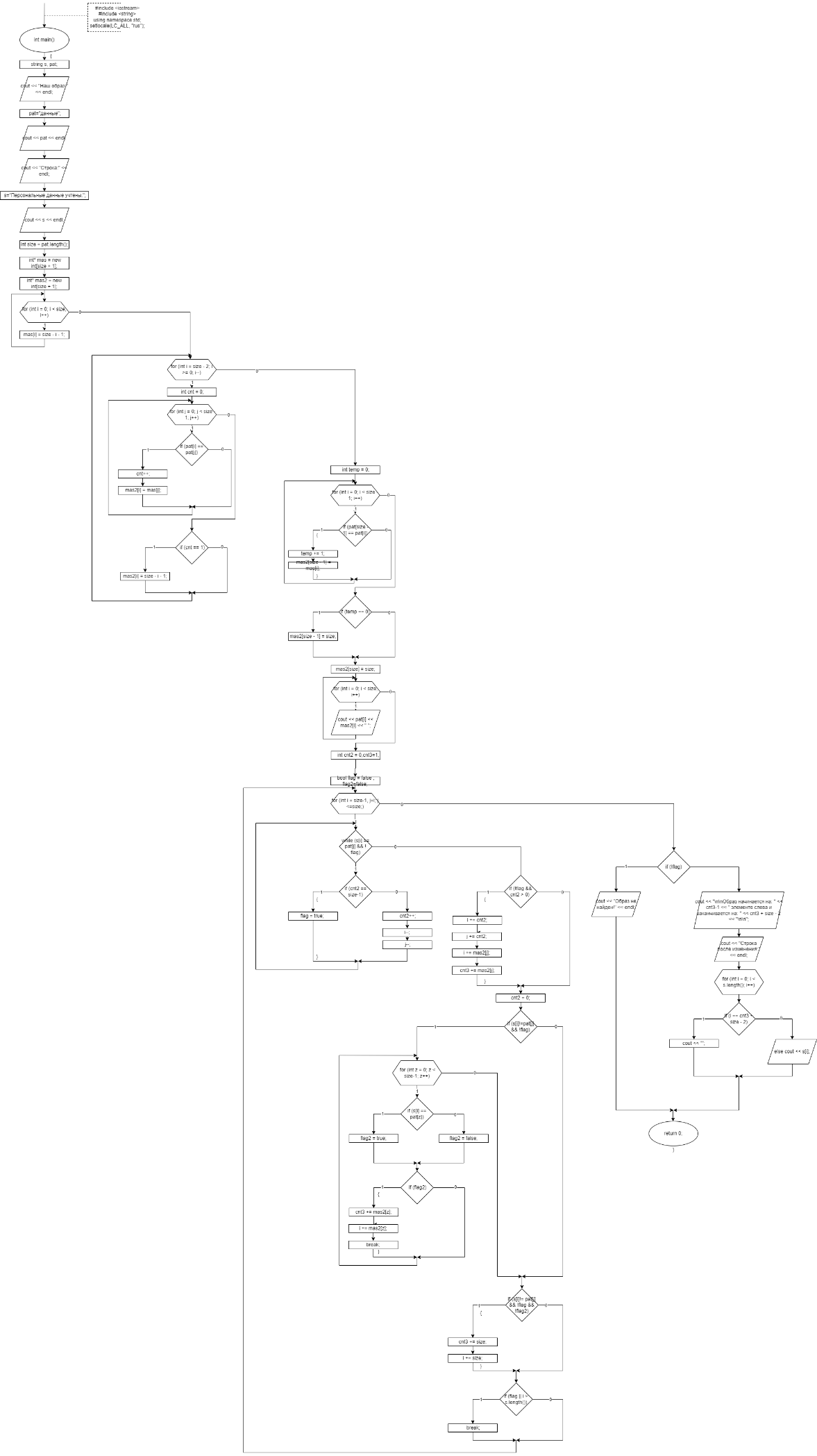
Найти подстроку strl, вывести индекс первого символа подстроки в строке

Анализ задачи:

Алгоритм Кнута-Морриса-Пратта используется для поиска подстроки (образца) в строке. Двигаемся по строке и сравниваем последовательно символы с образцом. Не совпало, перемещаем начало сравнения на один шаг и снова сравниваем. И так до тех пор, пока не найдем образец или не достигнем конца строки.

1. Вводится образ и строка, в которой он должен найтись;
2. Префикс-функция для i-го символа образа возвращает значение, равное максимальной длине совпадающих префикса и суффикса подстроки в образе, которая заканчивается i-м символом. Это значение будем хранить в d[i];
3. Для заполнения d[i] идём методом суффикса и префикса, сравнивая все их варианты по длине. Если они равны некоторой длине, то записываем максимальную длину в d[i];
4. Так продолжаем до конца образа, определяя d[i];
5. Если после сравнения значение j совпадает с размером образа, то образ найден, иначе не найден.

**Блок схема:**

****

**Код программы:**

#include <iostream>

#include <string>

using namespace std;

int KnutSearch(string& str, string& pat)

{

int strsize, patsize;

strsize = str.size();

patsize = pat.size();

if (strsize == 0) cout << "Строка пустая! ОШИБКА!" << endl;

else if (patsize == 0) cout << "Образ пуст! ОШИБКА!" << endl;

else

{

int i = 0, j = -1;

int\* d = new int[100];

d[0] = -1;

while (i < patsize - 1)

{

while (j >= 0 && pat[i] != pat[j]) j = d[j];

i++;

j++;

if (pat[i] == pat[j]) d[i] = d[j];

else d[i] = d[j];

}

i = 0, j = 0;

while ((j < patsize) && (i < strsize))

{

while (j >= 0 && str[i] != pat[j]) j = d[j];

i++;

j++;

}

delete[] d;

if (j == patsize) return i - j;

else return -1;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

string str, pat;

cout << "Введите строку:" << endl;

getline(cin, str);

cout << "Введите образ поиска:" << endl;

cin >> pat;

int index;

index = KnutSearch(str, pat);

if (KnutSearch(str, pat) != -1) cout << "Индекс первого символа в подстроки в строке: " << index << endl;

else cout << "Образ не найден!";

return 0;

}