Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

**Лабораторная работа № 17**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**(Семестр** 2)

Выполнил студент гр. ИВТ-21-1б

Гребнев Алексей Дмитриевич

Проверил:

Яруллин Денис Владимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г.Пермь -2022

**Постановка задачи**

Для 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).

**Исходные данные**

Вариант 10:  
Найти подстроку strl длиной k символов и заменить ее на подстроку, следующую после нее длиной k символов (например, подстрока ab в строке cvabghuj заменится: cvghghuj)

#include <iostream>

#include <string>

using namespace std;

int KMP\_search(string str, string templ)

{

int i, j, strlen, templlen;

strlen = str.size();

templlen = templ.size();

int\* d = new int[templlen];

//Вычисление префикс-функции

i = 0;

j = -1;

d[0] = -1;

while (i < templlen - 1)

{

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

{

j = d[j];

}

++i;

++j;

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

{

d[i] = d[j];

}

else

{

d[i] = j;

}

}

//Поиск строки

for (i = 0, j = 0; (i <= strlen - 1) && (j <= templlen - 1); i++, j++)

{

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

{

j = d[j];

}

}

delete[] d;

//Если совпадения есть, то возвращаем позицию

//Если нет, вывод -1

if (j == templlen)

{

return i - j;

}

else

{

return -1;

}

}

int main()

{

setlocale(LC\_ALL, "RUS");

string str, templ;

cout << "Введите строку, в которой будем искать: ";

cin >> str;

cout << "Введите образец того, что мы должны найти: ";

cin >> templ;

int strlen = str.size();

int k = templ.size();

string newstr;

string tmp2str;

int index;

int j = 0;

index = KMP\_search(str, templ);

int pos = index + k;

if (KMP\_search(str, templ) != -1)

{

cout << "Заданный образец в строке находится под индексом => " << index << " !" << endl;

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

{

tmp2str[j] = str[i];

j++;

}

newstr = str;

for (int i = index; i < strlen; i++)

{

newstr[i] = str[pos];

pos++;

}

}

else

cout << "Заданный образец не найден в строке! " << endl;

cout << "Изменённая строка (newstr): " << endl;

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

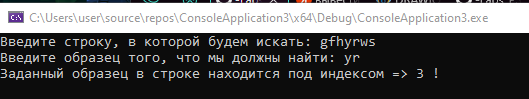
{

cout << newstr[i] << " ";

}

}

**Результат**



**Блок-схема**