Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:

Для линейного и интерполяционного поиска: Дан массив целых чисел (или символов, если это дано по заданию), выполнить задание из своего варианта, используя указанный метод поиска. Массив можно заполнить вручную или случайными числами, размерность массива любая (не меньше 5). Массив для интерполяционного поиска предварительно отсортировать.

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).

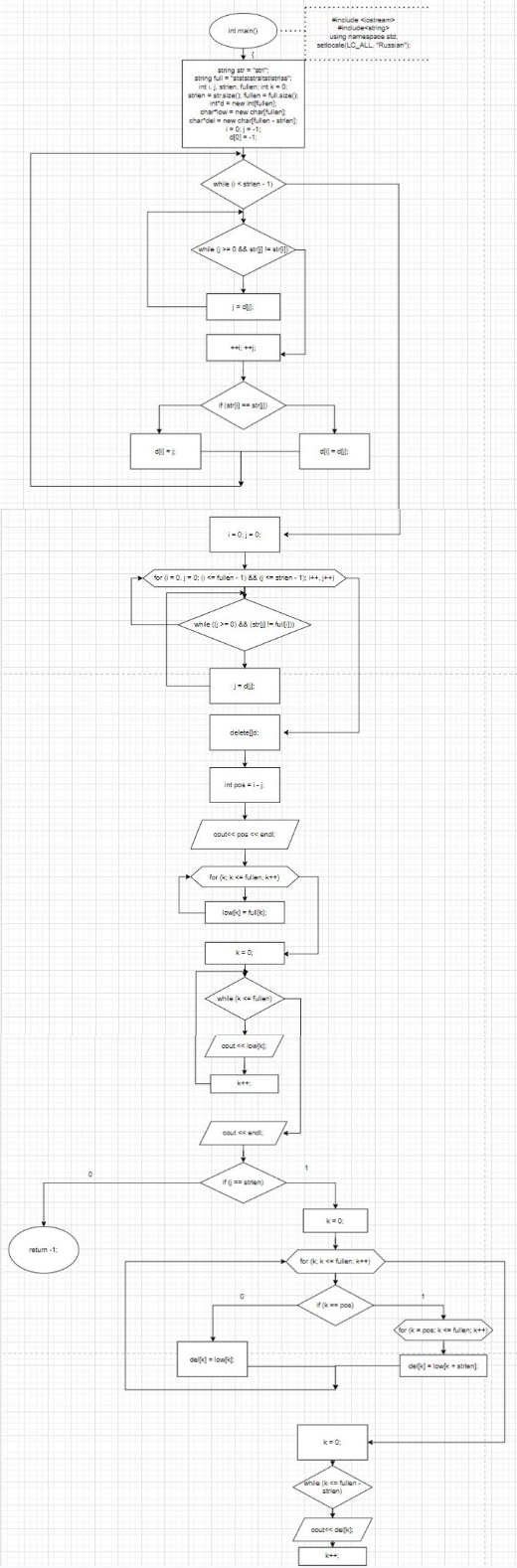
Анализ задачи:

Инициализируем и обьявляем переменные-

1. i, j, k – счетчики
2. strlen, fullen –длины строк
3. str, full – сами строки
4. d[], low[], del[] – динамические массивы

Ищем префикс функции через цикл while с условиями в этом цикле. Промежуточный массив заполняем всей исходной строкой. В новый массив запишем все символы до индекса искомого образа строки, и потом +длину образа. Таким образом произошло удаление искомого образа из строки.

Блок-схема:



Код программы:

#include<iostream>

#include<string>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

string str = "strl";

string full = "ststststrsltstlstrlss";

int i, j, strlen, fullen; int k = 0;

strlen = str.size(); fullen = full.size();

int\*d = new int[fullen];

char\*low = new char[fullen];

char\*del = new char[fullen - strlen];

i = 0; j = -1;

d[0] = -1;

while (i < strlen - 1)

{

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

{

j = d[j];

}

++i; ++j;

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

{

d[i] = d[j];

}

else

{

d[i] = j;

}

}

i = 0; j = 0;

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

{

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

{

j = d[j];

}

}

delete[]d;

int pos = i - j;

cout<< pos << endl;

for (k; k <= fullen; k++)

{

low[k] = full[k];

}

k = 0;

while (k <= fullen)

{

cout << low[k];

k++;

}

cout << endl;

if (j == strlen)

{

k = 0;

for (k; k <= fullen; k++)

{

if (k == pos)

{

for (k = pos; k <= fullen; k++)

{

del[k] = low[k + strlen];

}

}

else if (k != pos)

{

del[k] = low[k];

}

}

k = 0;

while (k <= fullen - strlen)

{

cout<< del[k];

k++;

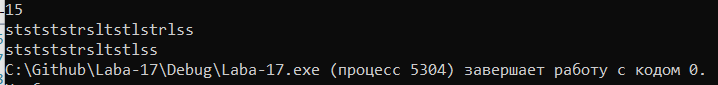
}

}

else { return -1; }

}

Скриншоты результатов:



Анализ результатов:

Программа работает корректно. Выводится строка с удаленным первым вхождением образа подстроки.