Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: «Поиск Кнута-Морриса-Пратта»

Семестр 2

Выполнил работу

Студент группы ИВТ-22-2б

Мифтахов Марат Ринатович

Проверил

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Г. Пермь-2023

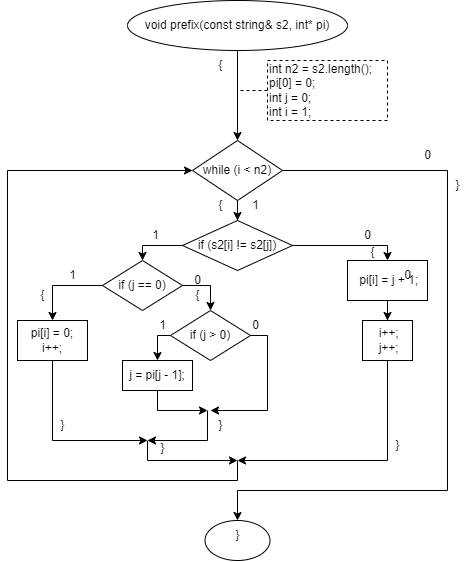
**Постановка задачи**

Реализовать код, который будет искать образ в строке поиском Кнута-Морриса-Пратта.

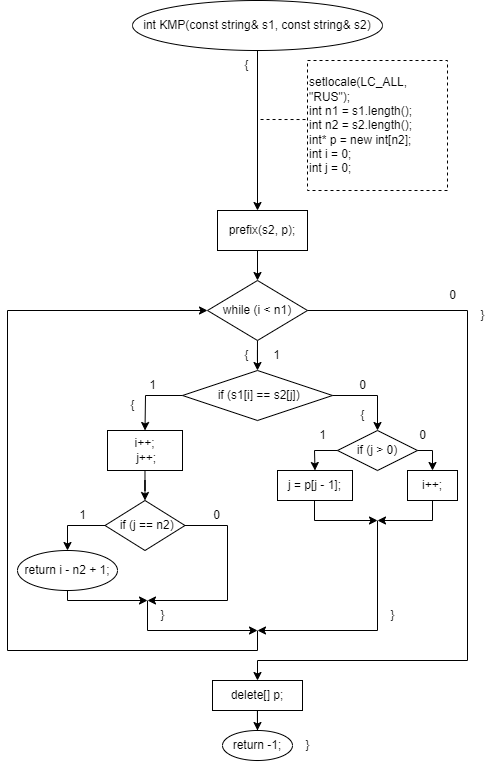
**Алгоритм решения задачи**

1. Создаётся функция void prefix() с параметрами const string& s2 (ссылка на подстроку) и int \*pi (указатель на на массив, в который будет записываться префикс функция). С помощью цикла каждому элементу массива присваивается число, равное расстоянию от первого элемента массива до любого другого. Если символы повторяются, то значение расстояния будет таким же, как в первый раз.
2. Создаётся функция int kmp () с параметрами const string& s1 (ссылка на строку) и const string& s2 (ссылка на подстроку). В функции создаётся массив int\* p, в котором будет находиться префикс, а затем вызывается функция заполнения этого массива. Далее создаётся цикл, который работает до тех пор, пока количество проверенных символов не достигнет значения длины строки. В цикле идёт несколько проверок: если символ строки и символ образа совпадают, то проверяются следующие символы; если они не совпадают, то индекс элемента строки увеличивается в на значение, соответствующее префикс-функции. Если количество совпавших символов совпадёт с длиной подстроки, функция вернёт номер первого совпавшего символа, если нет - вернёт -1.
3. В int main() создаётся две переменных - string s1 и string s2 (строка и подстрока соответственно). Затем пользователя просят ввести строку и подстроку. Вызывается функция поиска с параметрами s1 и s2. Если функция возвращает -1, то выводится сообщение о том, что подстрока не была найдена. Если любое другое значение - то оно выводится как номер вхождения первого символа подстроки в строку.

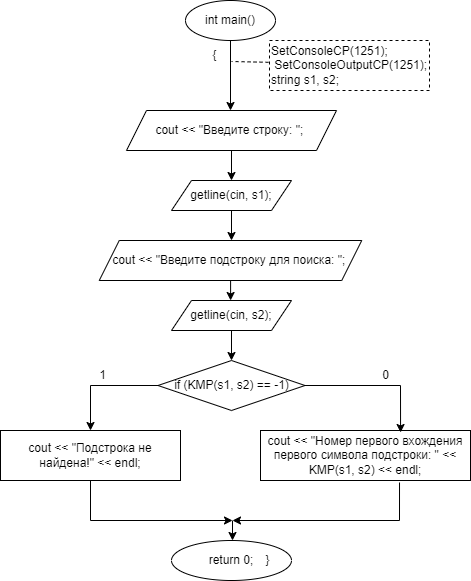
**Блок-схема**

****

**Рис. 1 – Блок-схема функции void prefix()**



**Рис. 2 – Блок-схема функции int KMP()**

****

**Рис. 3 – Блок-схема int main()**

**Код программы**

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

void prefix(const string& s2, int\* pi) //Префикс-функция

{

int n2 = s2.length();

pi[0] = 0;

int j = 0;

int i = 1;

while (i < n2)

{

if (s2[i] != s2[j]) //Проверка на несовпадение символов строки

{

if (j == 0) // Первому элементу префиксной функции присваивается значение 0

{

pi[i] = 0;

i++;

}

else

{

if (j > 0) j = pi[j - 1]; // индексу j присваевается значение предыдущего элемента префиксной функции

}

}

else // Если символы совпадают

{

pi[i] = j + 1; // Присваивается следующее по величине значение (первый элемент - 0, следующий 1..)

i++;

j++;

}

}

}

int KMP(const string& s1, const string& s2)

{

setlocale(LC\_ALL, "RUS");

int n1 = s1.length();

int n2 = s2.length();

int\* p = new int[n2];

prefix(s2, p);

int i = 0;

int j = 0;

while (i < n1) // Идёт до конца строки

{

if (s1[i] == s2[j]) // проверка элементов строки и образа на совпадение

{

i++;

j++;

if (j == n2) return i - n2 + 1; // Если количество совпавших символов равняется длине подстроки, выводится сообщение

}

else

{

if (j > 0) j = p[j - 1]; // Если встречается несовпадение, то указатель перемещается в соответствии с префикс-функцией

else

{

i++; // Если не совпал самый первый элемент, указатель смещвается на один вправо по строке

}

}

}

return -1;

delete[] p;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string s1, s2;

cout << "Введите строку: ";

getline(cin, s1);

cout << "Введите подстроку для поиска: ";

getline(cin, s2);

if (KMP(s1, s2) == -1) cout << "Подстрока не найдена!" << endl;

else cout << "Номер первого вхождения первого символа подстроки: " << KMP(s1, s2) << endl;

return 0;

}

**Результаты работы программы**





**Рис. 3 – Вывод консоли программы**