**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЁТ**

**По практической работе №2**

**По дисциплине «Программирование»**

Студент гр. 3351 \_\_\_\_\_\_\_\_\_\_\_\_ Морозов А.А.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ Рыжов Н.Г.

Санкт-Петербург

2023

цель работы

Написать программу с использованием строк C-Style, которая: вводит задаваемые пользователем строку текста и две подстроки: подстроку1 и подстроку2, осуществляет подсчет всех вхождений подстроки1 в строку, замену в тексте всех вхождений подстроки1 на подстроку2, подсчет и вывод на экран количества замен и полученной строки. Допускается разный размер подстрок 1 и 2 с учетом ограничений на максимальную длину строки (длину строки следует задать «с запасом» и контролировать ее переполнение). Выполнить задание без использования класса string и операторов динамического выделения памяти.

**ТЕКСТ ПРОГРАММЫ**

#include <iostream>

#include <stdio.h>

#include <cmath>

#include <windows.h>

#include <conio.h>

using namespace std;

const int lgth = 100; //Фиксированная длина для всех "строк" в программе

int main() {

int i, k, counts, entry\_num = 0, maxsize\_len, Len\_Pos, change\_counts = 0;

char str[lgth], subline1[lgth], subline2[lgth], copy\_str[lgth];

setlocale(LC\_ALL, "rus");

cout << "Введите строку" << endl;

gets\_s(str); // Ввод форматируемой строки

cout << "Введите подстроку №1" << endl;

gets\_s(subline1); // Ввод 1 подстроки

cout << "Введите подстроку №2" << endl;

gets\_s(subline2); // Ввод 2 подстроки

while ((strlen(subline1) >= strlen(str)) or (strlen(subline2) >= strlen(str))) { // Проверка правильности ввода

cout << "Длина подстрок должна быть меньше длины строки. Введите новые подстроки в таком же порядке\n"; // Вывод ошибки

gets\_s(subline1); // Повторный ввод 1 подстроки

gets\_s(subline2); // Повторный ввод 2 подстроки

}

for (i = 0; i < strlen(str); i++) { // Подсчет всех вхождений 1 подстроки в форматируемую строку

char tmp[lgth] = ""; // Пустой массив для записи в него повторяющихся элементов

counts = 0; // Обнуление счетчика

if (str[i] == subline1[0]) { // Поиск в начале 1 подстроки

k = i; // Присваивание значения для счетчика внутреннего цикла

while (str[k] == subline1[counts]) { // Проверка сопадения последующих символов подстроки1 со строкой

tmp[counts] = str[k]; // Запись во временный массив всех совпадающих символов

k++;

counts++;

}

if (!strcmp(subline1, tmp)) { // Cравнение 1 подстроки и нового массива с совпадающими символами

entry\_num++; // Счётчик вхождений

}

}

}

char\* entry = strstr(str, subline1); //вхождение подстроки1 в основную строку

while (entry) { //замена всех вхождений подстроки1 на подстроку2

Len\_Pos = strlen(str) - strlen(entry); //разница длин основной строки и вхождений подстроки1 в основную строку

strcpy\_s(copy\_str, entry + strlen(subline1)); //Копирование в новую строку

str[Len\_Pos] = '\0';

strcat\_s(str, subline2); // Присоединение к основной строке второй строки

strcat\_s(str, copy\_str); // Присоединение к основной строке копированной строки

entry = strstr(str, subline1); // Изменение вхождений подстроки1 в основную подстроку

change\_counts++; // Счётчик количества изменений

}

cout << "Измененная строка: " << str << endl;

cout << "Количество замен: " << change\_counts << endl;

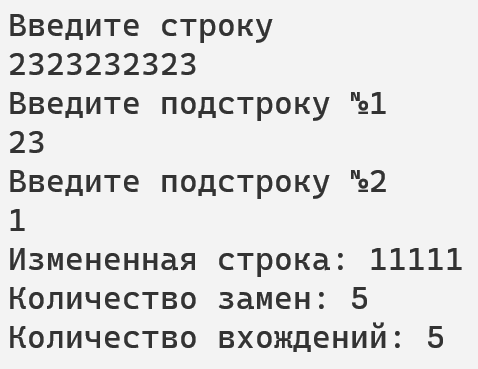
cout << "Количество вхождений: " << entry\_num << endl;

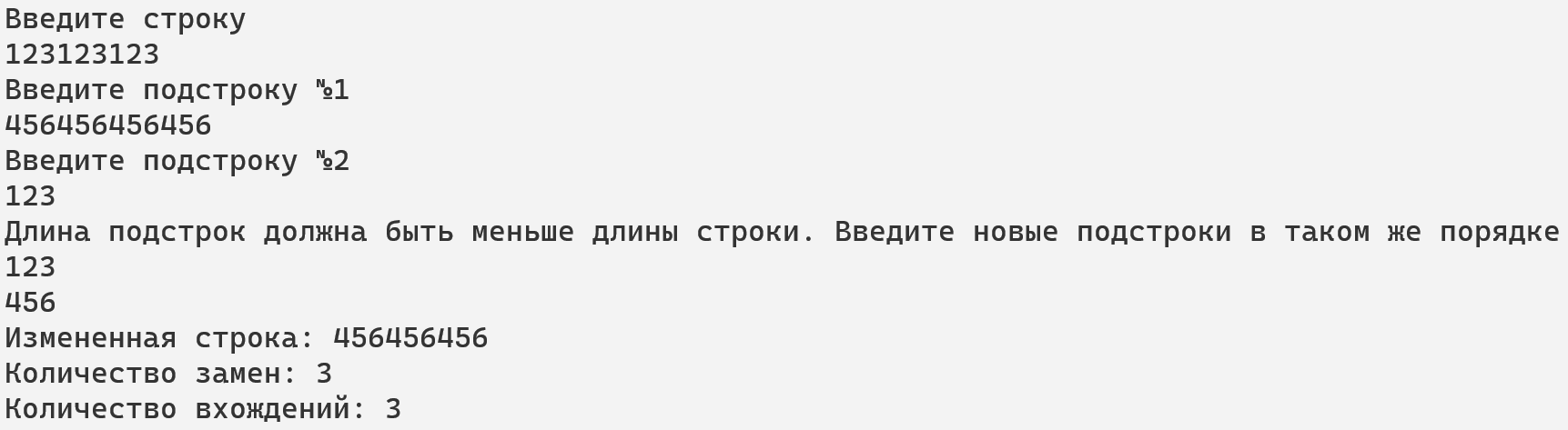
return 0;

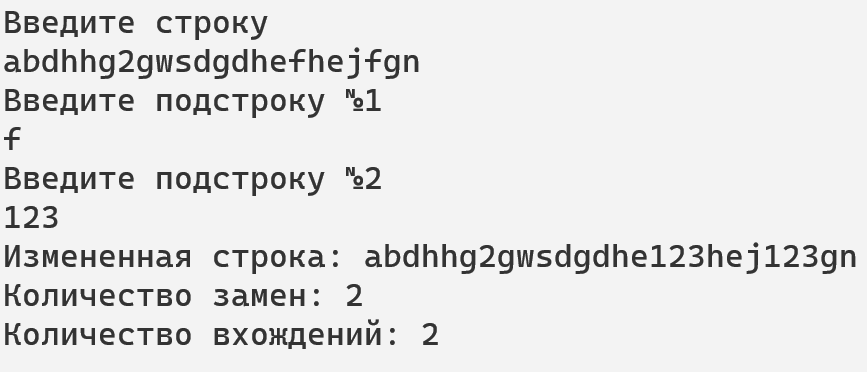
\_getch();

}

**Результаты выполнения программы**







**Пути дальнейшего улучшения программы**

1. Улучшение визуальной части программы
2. Увеличение длины форматируемой строки