

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

Институт информатики и кибернетики

Кафедра радиотехники

Отчет по индивидуальному заданию  
к лабораторной работе  
”Основы программирования на языке ассемблера  
процессоров фирмы INTEL”  
Вариант №24

Студент:	Согонов Е.А.
Преподаватель:	Корнилин Д.В.
Группа:	6364-120304D

Самара 2022

## СОДЕРЖАНИЕ

1	Формализация текстового задания . . . . .	3
2	Реализация требуемого функционала . . . . .	3

## 1. Формализация текстового задания

Исходное задание звучит так: *Замена в массиве заданного символа на цепочку из двух заданных символов*

Указания к выполнению задания следующие:

1. Все задания предполагают написание программы на языке ассемблера, представляющей собой ассемблерную вставку в Си-программе;
2. Под текстом понимается набор слов, состоящих из букв латиницы, цифр, точек, запятых, двоеточий, тире, разделенных пробелами;
3. Кодировка всех символов - ASCII;
4. Все текстовые строки заканчиваются двоичным нулем;

## 2. Реализация требуемого функционала

Был создан файл с кодом, показанным ниже.

```
1 //Замена в массиве заданного символа на цепочку из двух заданных символов
2 #include "windows.h"
3 #include "stdio.h"
4 char In[127], Out[256]; int N;
5 int APIENTRY WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance,
6 LPSTR lpCmdLine, int nCmdShow )
7 {
8     sprintf(In, "исходная строка"); //исходная строка
9     N=strlen(In); //длина строки
10    _asm //подключение ассемблерной вставки
11    { CLD ;сброс флага направления (DF), установка в 0.
12      ;при флаге DF=0 строки обрабатываются в сторону увеличения адресов
13      xor ecx,ecx ;сброс ecx
14      lea esi, In ;копирование в esi(регистра источника) адреса массива In
15      lea edi, Out ;копирование в edi(регистра приемника) адреса массива Out
16      mov al, 'a' ;Искомый символ
17
18      m_begin:
19          or al, 0 ;очистка флага zf
20          MOV ah, [esi] ;копируем байт из исходной строки в регистр ah
21          inc esi
22          inc ecx
23          cmp ah, al ;сравниваем байт с искомым
24          jz m_replace ;при совпадении с искомым переход к метке
25          ;при несовпадении продолжаем выполнять код
26          MOV [edi], ah ;копируем байт в приемник из регистра ah
27          inc edi ;инкрементируем индексный регистр на следующий байт
28          CMP ecx, N ;сравниваем значение ecx и длины исходной строки
29          jnz m_begin ;пока zf не равно 0, возвращаемся к метке начала
30          jz m_end ;когда zf равно 0 переходим к метке конца
31
32      m_replace:
33          mov byte ptr [edi], '=' ;записываем первый символ для замены
34          inc edi ;сдвигаемся дальше
35          mov byte ptr [edi], '+' ;записываем второй символ
36          inc edi ;
37          CMP ecx, N ;сравниваем значение ecx и длины исходной строки
38          jnz m_begin ;если строка не завершена, возвращаемся к началу(zf не ноль)
```

```

39                                     ;иначе продолжаем
40     m_end:
41     mov byte ptr [edi],0             ;дописываем завершающий нуль
42 }
43 MessageBox(NULL,Out,"Результат",MB_OK);
44 return 1;
45 }

```

Результат выполнения кода на рисунках 1,2.

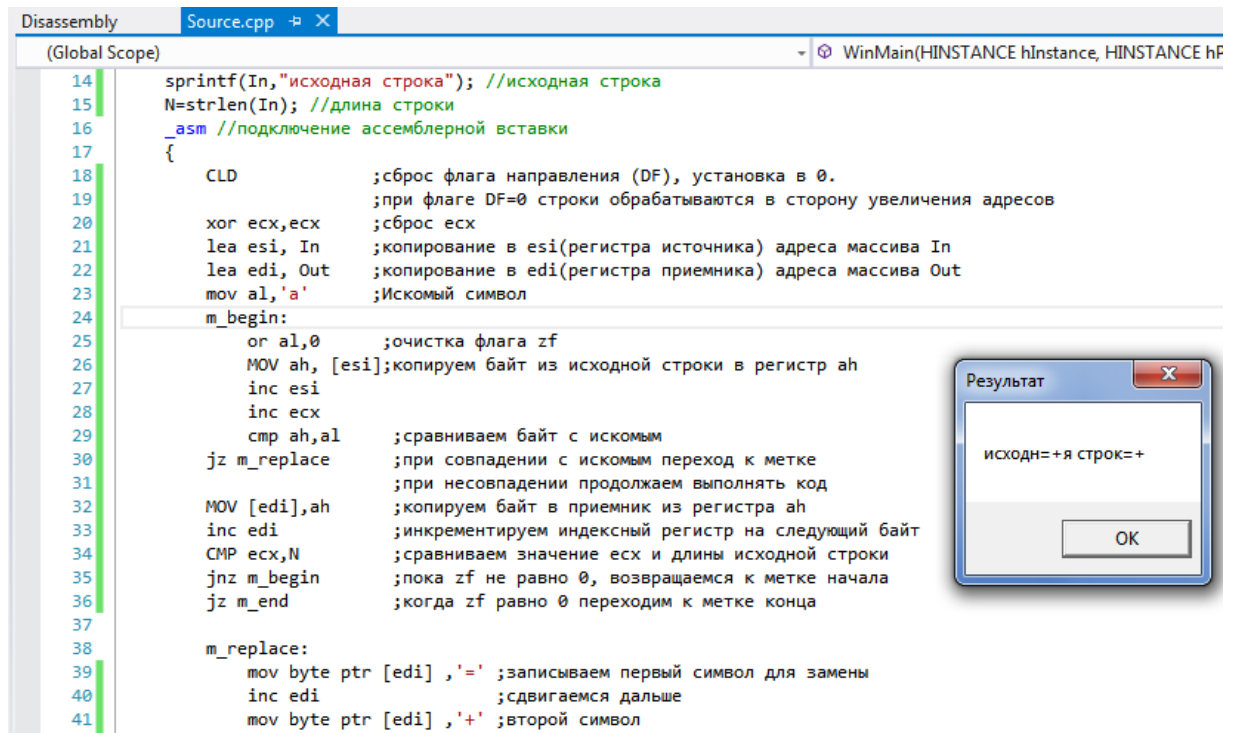


Рисунок 1 – Результат программы

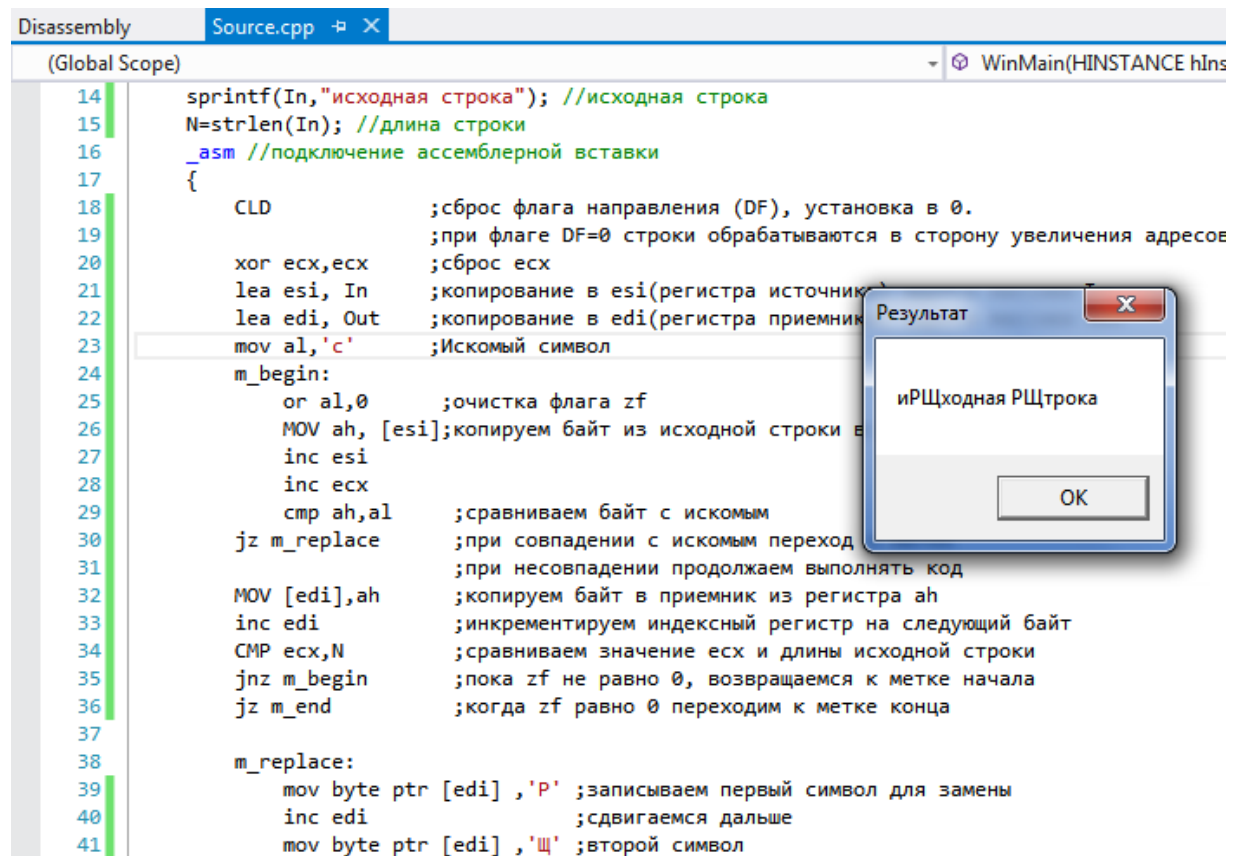


Рисунок 2 – Результат программы