**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №7  
дисциплина: Архитектура вычислительных систем  
тема: «Цепочечные команды»

Выполнил: ст. группы ВТ-31  
Ковалёв И. Д.  
Проверил: Осипов О. В.

**Белгород 2019**

**Цель работы:** изучение цепочечных команд для работы с одномерными массивами и строками

**Задания для выполнения к работе**

1. С клавиатуры вводятся слово a и строка. Исправить в строке одиночные ошибки в словах a. Считать, что длины исходного слова и исправляемого равны.

.486

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

INCLUDE C:\MASM32\INCLUDE\WINDOWS.INC

INCLUDE C:\MASM32\INCLUDE\KERNEL32.INC

INCLUDE C:\MASM32\INCLUDE\USER32.INC

INCLUDE C:\masm32\include\msvcrt.inc

INCLUDELIB C:\MASM32\LIB\USER32.LIB

INCLUDELIB C:\MASM32\LIB\KERNEL32.LIB

INCLUDELIB C:\masm32\lib\msvcrt.lib

.DATA

len dd 0

n dd 0

shift dd 0

inp\_str db 32 dup(?)

cmp\_word db 32 dup(?)

tmp\_str db 32 dup(?)

num\_format db "%d", 0

.CODE

get\_len proc

MOV EDI, [ESP + 4]

MOV AL, 0

MOV ECX, -1

REPNE SCASB

MOV ECX, EDI

SUB ECX, [ESP + 4]

RET 4

get\_len endp

;С клавиатуры вводятся слово a и строка. Исправить в строке одиночные

;ошибки в словах a. Считать, что длины исходного слова и исправляемого

;равны.

check proc

MOV ECX, n

MOV EBX, shift

comparing\_loop:

;Получаем очередное слово

PUSH EBX

CALL get\_next\_word

ADD EBX, len

INC EBX

MOV shift, EBX

PUSH offset cmp\_word

PUSH offset tmp\_str

call compare

CMP EAX, 1

JE skip\_word

call is\_one\_error

;CMP EAX, 1

;JL skip\_word

;XOR EAX, EAX

;CALL rewrite

MOV ECX, n

SUB ECX, shift

skip\_word:

LOOP comparing\_loop

check endp

;EAX = 0, если более одной ошибки в слове

is\_one\_error proc

PUSH ECX

PUSH EBX

PUSH EAX

XOR ECX, ECX

MOV ESI, offset cmp\_word

MOV EDI, offset tmp\_str

MOV ECX, len

INC ECX

XOR EAX, EAX; Счетчик количества ошибок

;zf, jz, jnz

REPE CMPSB

INC EAX

JNE ch1

ch2:

REPE CMPSB

JNE no\_edit

JMP edit

ch1:

CMP ECX, 0

JE edit

JMP ch2

edit:

;MOV EDI, ESI

MOV EAX, 2

no\_edit:

POP EAX

POP EBX

POP ECX

is\_one\_error endp

rewrite proc

;PUSH ECX

;POP ECX

rewrite endp

;EBX - количество пропущенных символов с начала строки

get\_next\_word proc

PUSH ECX

MOV ESI, offset inp\_str

ADD ESI, EBX; прибавляем количество символов, которые нужно пропустить

MOV EDI, offset tmp\_str

MOV ECX, len; сколько символов скопировать

REP MOVSB

POP ECX

RET 4

get\_next\_word endp

compare proc

PUSH ECX

PUSH EBX

PUSH ESI

PUSH EDI

MOV ECX, len

MOV ESI, [ESP + 4 \* 5]

MOV EDI, [ESP + 4 \* 6]

REPE CMPSB

JA j1

JB j1

JE j2 ; равны

j1:

MOV EAX, -1

JMP compare\_end

j2:

MOV EAX, 1

JMP compare\_end

compare\_end:

POP EDI

POP ESI

POP EBX

POP ECX

RET 8

compare endp

START:

PUSH offset inp\_str

CALL crt\_gets

ADD ESP, 4

PUSH offset inp\_str

CALL get\_len

MOV n, ECX

PUSH offset cmp\_word

CALL crt\_gets

ADD ESP, 4

PUSH offset cmp\_word

CALL get\_len

DEC ECX

MOV len, ECX

;XOR EAX, EAX

;MOV AX, WORD PTR n

;MOV BX, WORD PTR len

;DIV BX

call check

call crt\_\_getch

push NULL

call ExitProcess

END START

Результаты работы программы:



