**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО**

**ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**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

по теме Логические команды и команды сдвига

Выполнил: ст. группы ВТ-32  
Воскобойников И. С.

Проверил: Осипов.О.В

**Белгород 2020**

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

**Задания для выполнения к работе**

Написать программу для решения соответствующего варианта задания. По умолчанию, если в задании не оговорено, считать что в тексте могут быть только буквы русского алфавита, латинские буквы, цифры, пробелы и нулевой символ как признак окончания строки. Пробелов между словами может быть несколько. Необходимые операции копирования, сравнения, поиска и другие организовать в виде подпрограмм, используя цепочечные команды. Если в результате преобразования длина строки увеличивается, то изначально зарезервировать для неё большой объём памяти.

Найти самое часто встречающееся слово данной строки.

.686 ; Тип процессора

.model flat, stdcall ; Модель памяти и стиль вызова подпрограмм

option casemap: none ; Чувствительность к регистру

; --- Подключение файлов с кодом, макросами, константами, прототипами функций и т.д.

include c:\masm32\include\windows.inc

include c:\masm32\include\kernel32.inc

include c:\masm32\include\user32.inc

include c:\masm32\include\msvcrt.inc

; --- Подключаемые библиотеки ---

includelib c:\masm32\lib\user32.lib

includelib c:\masm32\lib\kernel32.lib

includelib c:\masm32\lib\msvcrt.lib

.DATA

strr DB 100 DUP (?)

wrd DB 100 DUP (?)

fPrintfStr DB "Word: ", 0

fPrintfCount DB "Count: %i", 0

.CODE

; length\_str (char\* s) -- Возвращает длину строки s в EAX без учета 0-символа

length\_str proc

PUSHF

PUSHAD

MOV EBX, [ESP + 38]

MOV EAX, 0h

MOV ECX, -1

MOV EDI, EBX

REPNE SCASB

SUB EDI, EBX

DEC EDI

MOV [ESP + 28], EDI

POPAD

POPF

RET 4

length\_str endp

; take\_word\_with\_position (char\* s, char\* w, unsigned\* i) -- Считывает из строки s первое найденное, начиная с позиции i, слово в w; i := позиция в s за словом w

take\_word\_with\_position proc;

PUSHF

PUSHAD

PUSH [ESP + 38]

CALL length\_str

MOV EBX, EAX

MOV EAX, ' '

MOV EBP, [ESP + 46]

MOV EDI, [EBP]

MOV ECX, EBX

SUB ECX, EDI

CMP ECX, 0h

JLE twwp\_end

ADD EDI, [ESP + 38]

REPE SCASB

DEC EDI

MOV ESI, EDI

SUB ESI, DWORD PTR [ESP + 38]

SUB ESI, [EBP]

ADD [EBP], ESI

INC ECX

CMP BYTE PTR [EDI], ' '

JE twwp\_spaces

MOV ESI, EDI

REPNE SCASB

DEC EDI;

CMP BYTE PTR [EDI], ' '

JNE twwp\_no\_space

JMP twwp\_copy

twwp\_no\_space:

INC EDI;

JMP twwp\_copy

twwp\_copy:

MOV ECX, EDI

SUB ECX, ESI

ADD [EBP], ECX

MOV EDI, [ESP + 42]

REP MOVSB;

MOV BYTE PTR [EDI], 0h

JMP twwp\_end

twwp\_spaces:

INC DWORD PTR [EBP]

JMP twwp\_end

twwp\_end:

POPAD

POPF

RET 12

take\_word\_with\_position endp

; str\_equal (char\* s1, char\* s2) -- Возвращает в EAX 1, если s1 = s2, или 0, если s1 != s2

str\_equal proc

PUSHF

PUSHAD

PUSH [ESP + 38]

CALL length\_str

MOV ECX, EAX

PUSH [ESP + 42]

CALL length\_str

CMP ECX, EAX

JNZ se\_res\_0

MOV ESI, [ESP + 38]

MOV EDI, [ESP + 42]

REPE CMPSB;

JNZ se\_res\_0

MOV DWORD PTR [ESP + 28], 1

JMP se\_end

se\_res\_0:

MOV DWORD PTR [ESP + 28], 0

JMP se\_end

se\_end:

POPAD

POPF

RET 8

str\_equal endp

; count\_word\_in\_str (char\* s, char\* w) -- Возвращает в EAX кол-во слов w в строке s

count\_word\_in\_str proc

PUSHF

PUSHAD

PUSH [ESP + 38]

CALL length\_str

MOV EDX, EAX

SUB ESP, 4h

MOV DWORD PTR [ESP], 0h

MOV EBP, ESP

MOV EBX, 0h

SUB ESP, 50

cwis\_again:

PUSH EBP

SUB EBP, 50

PUSH EBP;

PUSH DWORD PTR [EBP + 92]

CALL take\_word\_with\_position

PUSH EBP

ADD EBP, 50

PUSH [EBP + 46]

CALL str\_equal

CMP EAX, 0h

JZ cwis\_again\_continue

INC EBX

cwis\_again\_continue:

CMP DWORD PTR [EBP], EDX

JNZ cwis\_again

ADD ESP, 54

MOV DWORD PTR [ESP + 28], EBX

POPAD

POPF

RET 8

count\_word\_in\_str endp

; count\_max\_word (char\* s, char\* w) -- Записывает в w самое часто встречающееся слово в строке s. Возвращает в EAX кол-во его экземпляров в строке

count\_max\_word proc

PUSHF

PUSHAD

PUSH [ESP + 38]

CALL length\_str

MOV EDX, EAX

SUB ESP, 4h

MOV DWORD PTR [ESP], 0h

MOV EBP, ESP

SUB ESP, 50

MOV EBX, 0h

cmw\_again:

PUSH EBP

SUB EBP, 50

PUSH EBP

PUSH [EBP + 92]

CALL take\_word\_with\_position

PUSH EBP

PUSH [EBP + 92]

CALL count\_word\_in\_str

CMP EBX, EAX

JL cmw\_again\_new\_word

JMP cmw\_again\_continue

cmw\_again\_new\_word:

MOV EBX, EAX

PUSH EBP

CALL length\_str

MOV ECX, EAX

INC ECX

MOV ESI, EBP

MOV EDI, [EBP + 96]

REP MOVSB

JMP cmw\_again\_continue

cmw\_again\_continue:

ADD EBP, 50

CMP DWORD PTR [EBP], EDX

JNZ cmw\_again

ADD ESP, 54

MOV DWORD PTR [ESP + 28], EBX

POPAD

POPF

RET 8

count\_max\_word endp

START:

PUSH OFFSET strr

CALL crt\_gets

ADD ESP, 4h

PUSH OFFSET wrd

PUSH OFFSET strr

CALL count\_max\_word

PUSH EAX

PUSH OFFSET fPrintfStr

CALL crt\_printf

ADD ESP, 4h

PUSH OFFSET wrd

CALL crt\_puts

ADD ESP, 4h

PUSH OFFSET fPrintfCount

CALL crt\_printf

ADD ESP, 8h

call crt\_\_getch

push 0

call ExitProcess

END START