Федеральное агентство связи

Ордена трудового Красного Знамени федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Лабораторная работа № 1

«Технология работы с программами на языке Ассемблера. Элементарные операции.»

Выполнил: студент группы БВТ1904 Абакаров Г. Г.

Москва, 2020

**Цель работы:** Ознакомиться с технологией написания и отладки программ на языке ассемблера. Ознакомиться со структурой программы и директивами языка. Ознакомиться с операциями пересылки данных, передачи управления и прерываниями ввода-вывода.

**Порядок выполнения работы:** Написать программу на ассемблере, выполняющую операции над строкой символов согласно заданиям, приведенным ниже. Общие требования: -- выделить буфер для хранения 200 символов; -- строку символов ввести с клавиатуры; -- ввод строки символов может быть завершен клавишей «Enter» или по заполнению буфера; -- дополнительный буфер для хранения промежуточных результатов обработки строки не вводить; -- константы задать с помощью директивы EQU; -- старт программы, ввод – вывод данных и обработку ошибок оформлять выводом в консоли поясняющих строк.

**Выполнение:**

1) Отсортировать символы в строке по значению ASCII кода символа.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0 *;*

**call** print *; вывод сообщения msg0*

**add** **sp**, 2 *;*

**cmp** **ax**, 5

**je** exit

**push** offset pass *;*

**push** 200 *;*

**push** offset buffer1 *; ввод в buffer1*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset task\_msg *;*

**call** print *; вывод сообщения task\_msg*

**add** **sp**, 2 *;*

**push** offset buffer1 *;*

**call** sort *; отсортировать символы по ANSII коду*

**add** **sp**, 2 *;*

**push** offset buffer1 *;*

**call** print *; вывод результата*

**add** **sp**, 2 *;*

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_l10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_l10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_l10 *; вернуться в .\_l10*

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_l10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_l10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

input endp

sort proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 4

a1 **equ** **bp**+4

buf1 **equ** **bp**-2

buf2 **equ** **bp**-4

*; Отсортировать символы в строке по значению ASCII кода символа.*

**push** **[**a1**]**

**call** strlen *; сначала нужно найти длину строки*

**add** **sp**, 2

**mov** **[**buf2**]**, **ax** *; ax - длина строки, возвращаемое функцией strlen*

**mov** **[**buf1**]**, **word** ptr 1 *;*

L1:

**mov** **ax**, **[**buf1**]**

**mov** **bx**, **[**buf2**]**

**cmp** **ax**, **bx**

**jge** L2

**cmp** **ax**, 0

**je** L3

**mov** **cx**, **[**a1**]** *; cx -*

**add** **cx**, **[**buf1**]** *;*

**mov** **di**, **cx** *;*

**mov** **dh**, **[di]** *;*

**dec** **cx**

**mov** **di**, **cx**

**mov** **dl**, **[di]**

**cmp** **dl**, **dh**

**jle** L3

**mov** **di**, **cx**

**mov** **[di]**, **dh**

**inc** **cx**

**mov** **di**, **cx**

**mov** **[di]**, **dl**

**dec** **word** ptr **[**buf1**]**

**jmp** L1

L3:

**inc** **word** ptr **[**buf1**]**

**jmp** L1

L2:

**add** **sp**, 4

**pop** **bp**

**ret**

sort endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]** *;*

**call** strlen *;*

**add** **sp**, 2 *;*

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

*; cтроки в данной программе*

*; заканчиваются с нулевым символом*

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

**.data**

buffer1 **db** 201 dup**(**0**)**

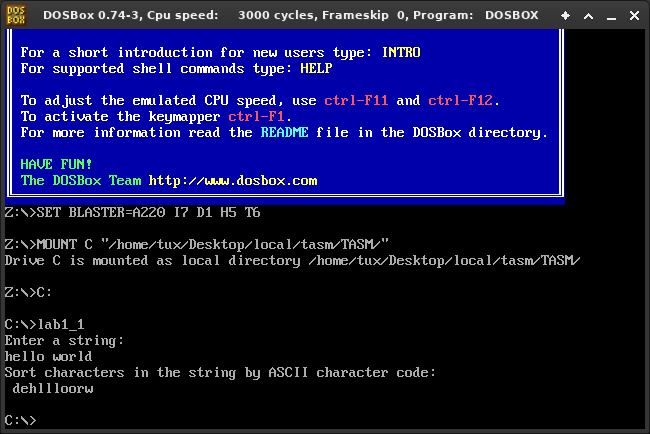
buffer2 **db** 201 dup**(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

task\_msg **db** 'Sort characters in the string by ASCII character code: ', 10, 0

end main

Пример работы программы:



2) Выполнить реверс слов строки.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0

**call** print

**add** **sp**, 2

**cmp** **ax**, 5

**je** exit

**push** offset pass

**push** 200

**push** offset buffer1

**call** input

**add** **sp**, 6

**push** offset task\_msg

**call** print

**add** **sp**, 2

**push** offset buffer1

**call** reverse

**add** **sp**, 2

**push** offset buffer1

**call** print

**add** **sp**, 2

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

reverse proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 6

a1 **equ** **bp**+4

buf4 **equ** **bp**-2

buf5 **equ** **bp**-4

buf6 **equ** **bp**-6

**mov** **cx**, **[**a1**]**

**mov** **[**buf4**]**, **cx**

L11:

**mov** **cx**, **[**buf4**]**

**mov** **[**buf5**]**, **cx**

L6:

**mov** **ax**, **[**buf5**]**

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**inc** **[**buf5**]**

**cmp** **dh**, 32

**je** L6

**cmp** **dh**, 0

**je** L10

**mov** **ax**, **[**buf5**]**

**mov** **[**buf6**]**, **ax**

**dec** **[**buf6**]**

L7:

**mov** **ax**, **[**buf5**]**

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**inc** **[**buf5**]**

**cmp** **dh**, 0

**je** L8

**cmp** **dh**, 32

**jne** L7

L8:

**mov** **ax**, **[**buf5**]**

**mov** **[**buf4**]**, **ax**

**sub** **[**buf5**]**, 2

L9:

**mov** **ax**, **[**buf5**]**

**mov** **bx**, **[**buf6**]**

**cmp** **ax**, **bx**

**jle** L11

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **dl**, **[di]**

**mov** **di**, **ax**

**mov** **[di]**, **dl**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**dec** **[**buf5**]**

**inc** **[**buf6**]**

**jmp** L9

L10:

**add** **sp**, 6

**pop** **bp**

**ret**

reverse endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_l10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_l10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_l10 *; вернуться в .\_l10*

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_l10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_l10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

input endp

**.data**

buffer1 **db** 201 dup**(**0**)**

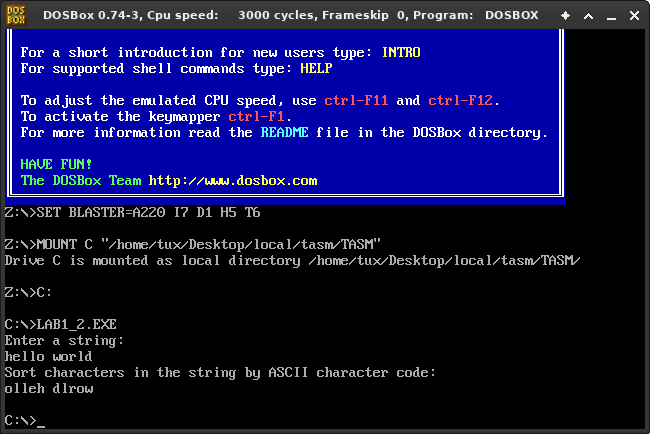
buffer2 **db** 201 dup**(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

task\_msg **db** 'Sort characters in the string by ASCII character code: ', 10, 0

end main

Пример работы программы



3) Выполнить реверс символов слов строки.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0

**call** print

**add** **sp**, 2

**cmp** **ax**, 5

**je** exit

**push** offset pass

**push** 200

**push** offset buffer1

**call** input

**add** **sp**, 6

**push** offset task\_msg

**call** print

**add** **sp**, 2

**push** offset buffer1

**call** reverse\_symbols

**add** **sp**, 2

**push** offset buffer1

**call** print

**add** **sp**, 2

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_l10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_l10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_l10 *; вернуться в .\_l10*

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_l10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_l10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

input endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

reverse\_symbols proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**dec** **[**a1**]**

L11:

**inc** **[**a1**]**

**mov** **si**, **[**a1**]**

**mov** **bl**, **[si]**

**mov** **al**, 0

**cmp** **bl**, 0 *; если это конец строки,*

**je** L10 *; то завершить подпрограмму*

**cmp** **bl**, 32 *; если символ - пробел,*

**je** L11 *; то пропустить*

**mov** **cx**, 8

L9:

**dec** **cx**

**shr** **al**, 1

**shl** **bl**, 1

**jc** L8

**inc** **al**

L8:

**cmp** **cx**, 0

**je** L9

**mov** **si**, **[**a1**]**

**mov** **[si]**, **al**

**jmp** L11

L10:

**pop** **bp**

**ret**

reverse\_symbols endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

**.data**

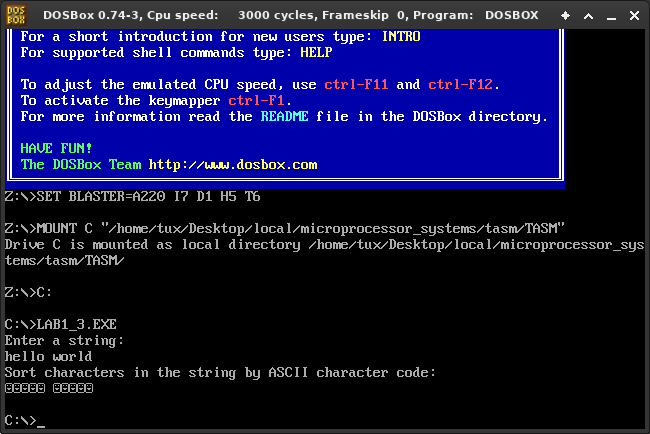
buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

task\_msg **db** 'Sort characters in the string by ASCII character code: ', 10, 0

end main

 Пример работы программы

4) Отсортировать слова в строке по значению ASCII кодов символов.

Код программы:

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0

**call** print

**add** **sp**, 2

**push** offset pass

**push** 200

**push** offset buffer1

**call** input

**add** **sp**, 6

**push** offset task\_msg

**call** print

**add** **sp**, 2

**push** offset buffer3

**push** offset buffer2

**push** offset buffer1

**call** sort\_words

**add** **sp**, 6

**push** offset buffer2

**call** print

**add** **sp**, 2

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

**ret**

main endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

split proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**sub** **sp**, 4

buf1 **equ** **bp**-2

buf2 **equ** **bp**-4

**mov** **[**buf2**]**, 0

**mov** **ax**, **[**a2**]**

**add** **ax**, 2

**mov** **[**buf1**]**, **ax**

J1:

**push** **[**a1**]**

**call** find\_any\_word

**add** **sp**, 2

**cmp** **bx**, 0

**je** J2

**mov** **[**a1**]**, **ax**

**add** **[**a1**]**, **bx**

**mov** **di**, **[**buf1**]***;*

**mov** **[di]**, **ax**

**add** **di**, 2

**mov** **[di]**, **bx**

**inc** **word** ptr **[**buf2**]**

**add** **[**buf1**]**, 4

**jmp** J1

J2:

**mov** **di**, **[**a2**]**

**mov** **ax**, **[**buf2**]**

**mov** **[di]**, **ax**

**add** **sp**, 4

**pop** **bp**

**ret**

endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

sort\_words proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

**sub** **sp**, 16

buf5 **equ** **bp**-2

buf6 **equ** **bp**-4

buf7 **equ** **bp**-6

buf8 **equ** **bp**-8

buf9 **equ** **bp**-10

buf10 **equ** **bp**-12

buf11 **equ** **bp**-14

buf12 **equ** **bp**-16

*; функция сортирует слова по ANSII коду*

**push** **[**a3**]**

**push** **[**a1**]**

**call** split

**add** **sp**, 4

*; [buf7] - начало массива*

*; [buf6] - длина массива*

*; [buf5] - индекс в массиве*

**mov** **ax**, **[**a3**]**

**mov** **[**buf7**]**, **ax**

**add** **[**buf7**]**, **word** ptr 2

**mov** **di**, **ax**

**mov** **ax**, **[di]**

**mov** **cl**, 4

**mul** **cl**

**mov** **[**buf6**]**, **ax**

**mov** **[**buf5**]**, **word** ptr 4

L\_6:

**mov** **ax**, **[**buf5**]**

**mov** **bx**, **[**buf6**]**

**cmp** **ax**, **bx**

**jge** L\_71 *; здесь произошла ошибка 'relative jump out of range',*

*; поэтому jump разделил на 2 команды*

**cmp** **ax**, 0

**je** L\_81 *; здесь тоже произошла ошибка 'relative jump out of range'*

**mov** **cx**, **[**buf7**]**

**add** **cx**, **[**buf5**]**

*;сравнить строки*

**mov** **di**, **cx**

**mov** **dx**, **[di]**

**mov** **[**buf8**]**, **dx** *; [buf8] - указатель на первую строку*

**add** **cx**, 2

*;*

**mov** **di**, **cx**

**mov** **dx**, **[di]**

**mov** **[**buf9**]**, **dx** *; [buf9] - длина первой строки*

**sub** **cx**, 6

*;*

**mov** **di**, **cx**

**mov** **dx**, **[di]**

**mov** **[**buf10**]**, **dx** *; [buf10] - указатель на вторую строку*

**add** **cx**, 2

*;*

**mov** **di**, **cx**

**mov** **dx**, **[di]**

**mov** **[**buf11**]**, **dx** *; [buf11] - длина второй строки*

**sub** **cx**, 2

*;*

**mov** **[**buf12**]**, **cx**

*;*

**jmp** L\_72

L\_71: **jmp** L\_7

L\_72:

**jmp** L\_82

L\_81: **jmp** L\_8

L\_82:

**push** **[**buf11**]**

**push** **[**buf9**]**

**push** **[**buf10**]**

**push** **[**buf8**]**

**call** strcmpA

**add** **sp**, 8

**cmp** **ax**, 0

**je** L\_8

**cmp** **ax**, 1

**je** L\_8

*;*

**mov** **cx**, **[**buf12**]**

*;*

**mov** **dx**, **[**buf8**]** *; перестановка строк местами*

**mov** **di**, **cx**

**mov** **[di]**, **dx**

**add** **cx**, 2

*;*

**mov** **dx**, **[**buf9**]** *;*

**mov** **di**, **cx**

**mov** **[di]**, **dx**

**add** **cx**, 2

*;*

**mov** **dx**, **[**buf10**]** *;*

**mov** **di**, **cx**

**mov** **[di]**, **dx**

**add** **cx**, 2

*;*

**mov** **dx**, **[**buf11**]** *;*

**mov** **di**, **cx**

**mov** **[di]**, **dx**

**add** **cx**, 2

*;*

**sub** **[**buf5**]**, **word** ptr 4

**jmp** L\_6

L\_8:

**add** **[**buf5**]**, **word** ptr 4

**jmp** L\_6

L\_7:

*;;;*

**mov** **ax**, **[**a3**]**

**mov** **[**buf5**]**, **ax**

**add** **[**buf5**]**, **word** ptr 2

**mov** **di**, **ax**

**mov** **ax**, **[di]**

**mov** **[**buf6**]**, **ax**

**cmp** **ax**, 0

**je** V1

**jmp** V2

V:

**mov** **cx**, **[**a2**]**

**mov** **di**, **cx**

**mov** **[di]**, **byte** ptr 32

**inc** **[**a2**]**

V2:

**mov** **ax**, **[**buf5**]**

**mov** **di**, **ax**

**mov** **ax**, **[di]**

**add** **[**buf5**]**, **word** ptr 2

**mov** **bx**, **[**buf5**]**

**mov** **di**, **bx**

**mov** **bx**, **[di]**

**add** **[**buf5**]**, **word** ptr 2

**mov** **cx**, **[**a2**]**

**add** **[**a2**]**, **bx**

**push** **bx**

**push** **cx**

**push** **ax**

**call** copyA

**add** **sp**, 6

**dec** **[**buf6**]**

**mov** **ax**, **[**buf6**]**

**cmp** **ax**, 0

**jne** V

V1:

**mov** **cx**, **[**a2**]**

**mov** **di**, **cx**

**mov** **[di]**, **byte** ptr 0

**add** **sp**, 16

**pop** **bp**

**ret**

sort\_words endp

strcmpA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

a4 **equ** **bp**+10

*; a1 - первая строка*

*; а2 - вторая строка*

*; а3 - длина первой строки*

*; а4 - длина второй строки*

**mov** **ax**, **[**a3**]**

**mov** **bx**, **[**a4**]**

**cmp** **ax**, **bx**

**jle** \_l1

**mov** **ax**, **bx**

\_l1:

**mov** **bx**, **[**a1**]**

**mov** **cx**, **[**a2**]**

\_l2:

**cmp** **ax**, 0

**je** \_l4

**dec** **ax**

**mov** **di**, **bx** *;*

**mov** **dl**, **[di]**

**mov** **di**, **cx**

**mov** **dh**, **[di]**

**cmp** **dl**, **dh**

**je** \_l2

\_l3:

**cmp** **dl**, **dh**

**jg** \_l5

**jl** \_l6

\_l4:

**mov** **bx**, **[**a3**]**

**mov** **cx**, **[**a4**]**

**cmp** **bx**, **cx**

**jg** \_l5

**jl** \_l6

**mov** **ax**, 0

**pop** **bp**

**ret**

\_l5:

**mov** **ax**, 1

**pop** **bp**

**ret**

\_l6:

**mov** **ax**, -1

**pop** **bp**

**ret**

strcmpA endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16 *;*

**cmp** **dh**, 0 *;*

**je** .l17 *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

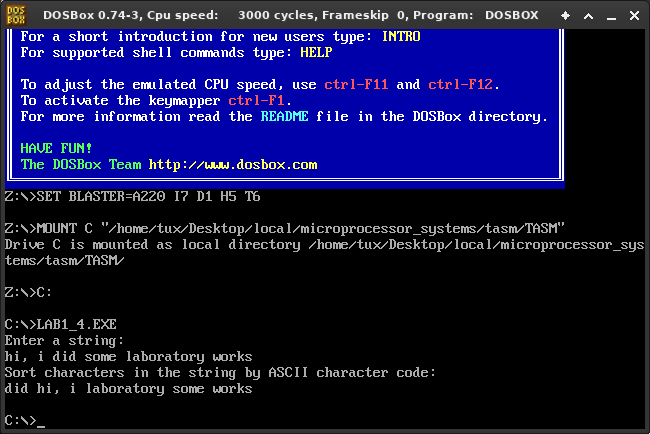
\_\_buf0 **dw** 201 dup **(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

task\_msg **db** 'Sort characters in the string by ASCII character code: ', 10, 0

end main

Пример работы программы



5) Удалить заданное слово (подстроку) в строке.

Код программы

**.model small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0 *;*

**call** print *; вывести строку msg0*

**add** **sp**, 2 *;*

**push** offset pass *;*

**push** 200 *;*

**push** offset buffer1 *; ввести строку в buffer1*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset msg1 *;*

**call** print *; вывести строку msg1*

**add** **sp**, 2 *;*

**push** offset is\_not\_space *;*

**push** 200 *;*

**push** offset buffer2 *; ввести слово в buffer2*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset buffer3 *;*

**push** offset buffer2 *; удалить слово buffer2*

**push** offset buffer1 *; из строки buffer1*

**call** delete\_word *; и записать результат в buffer3*

**add** **sp**, 6

**cmp** **ax**, -1 *; если строка не найдена, то*

**jne** .ll\_1 *; вывести ошибку*

*; иначе вывести результат выполнения*

**push** offset error\_message1 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer2 *;*

**call** print *; вывести ошибку и завершить*

**add** **sp**, 2 *;*

**push** offset error\_message2 *;*

**call** print *;*

**add** **sp**, 2 *;*

**jmp** exit

.ll\_1:

**push** offset task\_msg1 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer2 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset task\_msg2 *; вывести результат и завершить*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer3 *;*

**call** print *;*

**add** **sp**, 2 *;*

**call** newline *;*

exit:

**mov** **ax**,4C00h

**int** 21h

main endp

is\_not\_space proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, **[**a1**]**

**cmp** **al**, 32

**je** e2

**pop** **bp**

**ret**

e2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_not\_space endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

delete\_word proc

*; функция удаляет*

*; подстроку a1 из*

*; строки а2*

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 6

a1 **equ** **bp**+4

a2 **equ** **bp**+6

buf17 **equ** **bp**-2

buf18 **equ** **bp**-4

buf19 **equ** **bp**-6

**push** **[**a2**]**

**push** **[**a1**]**

**call** find

**add** **sp**, 4

**cmp** **ax**, -1

**je** \_L21

**mov** **[**buf17**]**, **ax**

**mov** **[**buf18**]**, **bx**

**mov** **ax**, **[**a3**]**

**mov** **[**buf19**]**, **ax**

**push** **[**buf17**]**

**push** **[**buf19**]**

**push** **[**a1**]**

**call** copyA

**add** **sp**, 6

**mov** **ax**, **[**a1**]**

**add** **ax**, **[**buf17**]**

**add** **ax**, **[**buf18**]**

**mov** **bx**, **[**buf19**]**

**add** **bx**, **[**buf17**]**

**push** **bx**

**push** **ax**

**call** copy

**add** **sp**, 4

**mov** **ax**, **[**buf19**]**

\_L21:

**add** **sp**, 6

**pop** **bp**

**ret**

delete\_word endp

copy proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**a2**]**

.l9:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**cmp** **dh**, 0

**jne** .l9

**pop** **bp**

**ret**

copy endp

find proc

*; функция ищет в строке а1 подстроку а2*

*; и возвращает индекс подстроки*

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

buf16 **equ** **bp**-2

**mov** **ax**, **[**a1**]**

**mov** **[**buf16**]**, **ax**

**dec** **[**buf16**]**

\_L17:

**inc** **[**buf16**]**

**mov** **ax**, **[**buf16**]**

**mov** **bx**, **[**a2**]**

\_L19:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **dl**, **[di]**

**inc** **ax**

**inc** **bx**

**cmp** **dl**, 0

**je** \_L20

**cmp** **dh**, 0

**je** \_L18

**cmp** **dh**, **dl**

**je** \_L19

**jne** \_L17

\_L20:

**mov** **bx**, **ax**

**sub** **bx**, **[**buf16**]**

**dec** **bx**

**mov** **ax**, **[**buf16**]**

**sub** **ax**, **[**a1**]**

**add** **sp**, 2

**pop** **bp**

**ret**

\_L18:

**mov** **ax**, -1

**add** **sp**, 2

**pop** **bp**

**ret**

find endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16 *;*

**cmp** **dh**, 0 *;*

**je** .l17 *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

\_\_buf0 **dw** 201 dup **(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

msg1 **db** 'Enter a word: ', 10, 0

task\_msg1 **db** 'Delete a word "',0

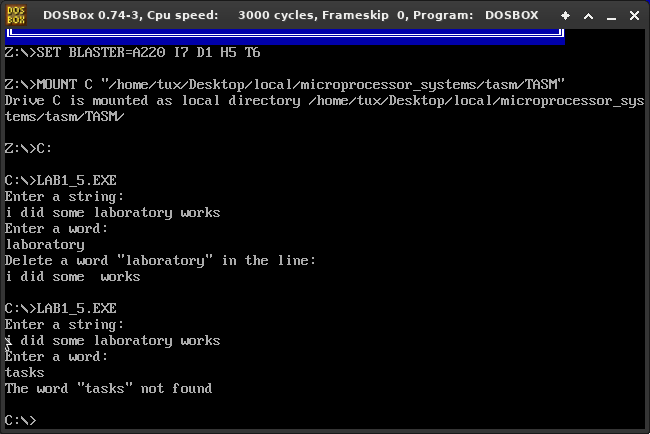
task\_msg2 **db** '" in the line: ', 10, 0

error\_message1 **db** 'The word "',0

error\_message2 **db** '" not found', 10, 0

end main

Пример работы программы



6) Заменить заданное слово (подстроку) в строке на другую заданную подстроку.

Код программы

.model small

.stack 100h

.code

main proc

mov ax, @data

mov ds, ax

push offset msg0

call print

add sp, 2

push offset pass

push 200

push offset buffer1

call input

add sp, 6

push offset msg1

call print

add sp, 2

push offset is\_not\_space

push 200

push offset buffer2

call input

add sp, 6

push offset msg1

call print

add sp, 2

push offset is\_not\_space

push 200

push offset buffer4

call input

add sp, 6

push offset buffer3

push offset buffer4

push offset buffer2

push offset buffer1

call replace\_word

add sp, 6

cmp ax, -1

jne .ll\_1

push offset error\_message1

call print

add sp, 2

push offset buffer2

call print

add sp, 2

push offset error\_message2

call print

add sp, 2

jmp exit

.ll\_1:

push offset task\_msg1

call print

add sp, 2

push offset buffer2

call print

add sp, 2

push offset task\_msg2

call print

add sp, 2

push offset buffer4

call print

add sp, 2

push offset task\_msg3

call print

add sp, 2

push offset buffer3

call print

add sp, 2

call newline

exit:

mov ax,4C00h

int 21h

main endp

is\_not\_space proc

push bp

mov bp, sp

a1 equ bp+4

mov ax, [a1]

cmp al, 32

je e2

pop bp

ret

e2:

mov ax, 0

pop bp

ret

is\_not\_space endp

newline proc

mov ah, 06h *;*

mov dl, 13 *; вывести новую линию*

int 21h *;*

mov ah, 06h *;*

mov dl, 10 *;*

int 21h *;*

newline endp

pass proc

push bp

mov bp, sp

mov ax, [bp+4]

pop bp

ret

pass endp

replace\_word proc

*; a1 - строка*

*; a2 - слово, которое нужно заменить*

*; a3 - слово, на которое нужно заменить*

*; a4 - буфер для новой строки*

push bp

mov bp, sp

sub sp, 12

a1 equ bp+4

a2 equ bp+6

a3 equ bp+8

a4 equ bp+10

buf20 equ bp-2

buf21 equ bp-4

buf22 equ bp-6

buf23 equ bp-8

buf24 equ bp-10

buf25 equ bp-12

push [a2]

push [a1]

call find

add sp, 4

cmp ax, -1

je \_L22

mov [buf20], ax

mov [buf21], bx

push [a3]

call strlen

add sp, 2

mov [buf22], ax

mov ax, [a4]

mov [buf23], ax

mov [buf24], ax

mov ax, [a1]

mov [buf25], ax

push [buf20]

push [buf23]

push [a1]

call copyA

add sp, 6

mov ax, [buf20]

add [buf24], ax

add [buf25], ax

push [buf24]

push [a3]

call copy

add sp, 4

mov ax, [buf22]

add [buf24], ax

mov ax, [buf21]

add [buf25], ax

push [buf24]

push [buf25]

call copy

add sp, 4

mov ax, [buf23]

\_L22:

add sp, 12

pop bp

ret

replace\_word endp

copyA proc

push bp

mov bp, sp

a1 equ bp+4

a2 equ bp+6

a3 equ bp+8

*; функция копирует а3 символов из а1 строки в а2 строку*

mov ax, [a1]*;ax - указатель на строку а1*

mov bx, [a2]*;bx - указатель на строку а2*

mov cx, [a3]

add cx, ax *;cx - указатель на конец строки а1*

.\_l9:

cmp ax, cx

je .\_l10

mov di, ax

mov dh, [di]

mov di, bx

mov [di], dh

inc ax

inc bx

jmp .\_l9

.\_l10:

pop bp

ret

copyA endp

copy proc

push bp

mov bp, sp

a1 equ bp+4

a2 equ bp+6

mov ax, [a1]

mov bx, [a2]

.l9:

mov di, ax

mov dh, [di]

mov di, bx

mov [di], dh

inc ax

inc bx

cmp dh, 0

jne .l9

pop bp

ret

copy endp

find proc

push bp

mov bp, sp

sub sp, 2

a1 equ bp+4

a2 equ bp+6

buf16 equ bp-2

*; функция ищет в строке а1 подстроку а2*

*; и возвращает индекс подстроки*

mov ax, [a1]

mov [buf16], ax

dec [buf16]

\_L17:

inc [buf16]

mov ax, [buf16]

mov bx, [a2]

\_L19:

mov di, ax

mov dh, [di]

mov di, bx

mov dl, [di]

inc ax

inc bx

cmp dl, 0

je \_L20

cmp dh, 0

je \_L18

cmp dh, dl

je \_L19

jne \_L17

\_L20:

mov bx, ax

sub bx, [buf16]

dec bx

mov ax, [buf16]

sub ax, [a1]

add sp, 2

pop bp

ret

\_L18:

mov ax, -1

add sp, 2

pop bp

ret

find endp

print proc

push bp

mov bp, sp

a1 equ bp+4

push [a1]

call strlen

add sp, 2

mov cx, ax

mov ah, 40h

mov dx, [a1]

mov bx, 1

int 21h

pop bp

ret

print endp

strlen proc

push bp

mov bp, sp

a1 equ bp+4

mov ax, 0

mov ax, [a1]

.l1:

mov di, ax

mov dl, [di]

inc ax

cmp dl, 0

jne .l1

sub ax, [a1]

dec ax

pop bp

ret

strlen endp

find\_any\_word proc

push bp

mov bp, sp

a1 equ bp+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

mov ax, [a1]

.l16: *;*

mov di, ax *;*

mov dh, [di] *;*

inc ax *;*

cmp dh, 32 *; пропустить пробелы*

je .l16 *;*

cmp dh, 0 *;*

je .l17 *;*

dec ax *;*

mov bx, ax *; ax - индекс слова в строке*

.l18:

inc bx *;*

mov di, bx *;*

mov dh, [di] *;*

cmp dh, 32 *;*

je .l19 *;*

cmp dh, 0 *;*

je .l19 *;*

jmp .l18

.l19:

sub bx, ax

pop bp

ret

.l17:

mov ax, 0

mov bx, 0

pop bp

ret

find\_any\_word endp

input proc

push bp

mov bp, sp

sub sp, 2 *; зарезервировать для новой переменной*

a1 equ bp+4 *; a1 - указатель на начало буффер*

a2 equ bp+6 *; a2 - длина буффера*

a3 equ bp+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 equ bp-2 *; новая переменная (указатель)*

mov ax, [a1]

mov [buf0], ax *; сохраняем значение а1 в buf0*

add [a2], ax *; помещаем в а2 указатель на конец буффера*

.\_ll10:

mov ah, 07h *; ввод*

int 21h

cmp al, 8 *; если нажата кнопка backspace, то*

je .\_l12 *; выполнить .\_l12*

cmp al, 13 *; если нажат enter, то*

je .\_l15 *; выполнить .\_l15*

push ax

call [a3] *; проверить символ на недопустимые значения*

add sp, 2

cmp al, 0 *; если символ недопустимый, то оно обращается в ноль*

je .\_ll10

mov si, [a1] *; записать символ*

mov [si], al *;*

mov ah, 06h

mov dl, al *; вывести символ*

int 21h

inc word ptr [a1] *; переместить указатель вперед*

mov ax, [a1] *;*

mov bx, [a2] *;*

cmp ax, bx *; если указатель в конце строки, то*

je .\_l14 *; выполнить .\_l14*

jmp .\_ll10 *; вернуться в .\_l10*

.\_l12:

mov ax, [a1]

mov bx, [buf0]

cmp ax, bx *; eсли указатель в начале строки, то*

je .\_ll10 *; вернуться в .\_l10*

dec word ptr [a1] *; cместить указатель назад*

mov ah, 06h *;*

mov dl, 8 *;*

int 21h *;*

mov ah, 06h *;*

mov dl, 32 *; cтереть символ*

int 21h *;*

mov ah, 06h *;*

mov dl, 8 *;*

int 21h *;*

jmp .\_ll10 *; вернуться в .\_l10*

.\_l14:

mov ah, 07h *;*

int 21h *; выполняется до тех пор,*

cmp al, 8 *; пока не нажата кнопка backspace*

je .\_l12 *; если нажата кнопка enter,*

cmp al, 13 *; то вывести новую линию*

je .\_l15 *; записать 0 в конец строки и завершить*

jmp .\_l14

.\_l15:

call newline *; вывести новую линию*

mov di, [a1]

mov [di], byte ptr 0 *; записать 0 в конец строки*

add sp, 2 *;*

pop bp *; завершить подпрограмму, освободить стек*

ret *;*

input endp

.data

buffer1 db 201 dup(0)

buffer2 db 201 dup(0)

buffer3 db 201 dup(0)

buffer4 db 201 dup(0)

\_\_buf0 dw 201 dup (0)

msg0 db 'Enter a string: ', 10, 0

msg1 db 'Enter a word: ', 10, 0

task\_msg1 db 'Replace the word "', 0

task\_msg2 db '" with the word "', 0

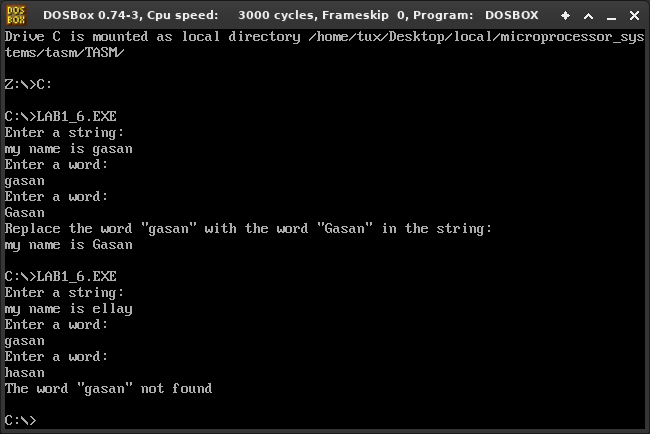
task\_msg3 db '" in the string: ', 10, 0

error\_message1 db 'The word "', 0

error\_message2 db '" not found', 10, 0

end main

Пример работы программы



7) Вставить в строке перед заданным словом другое заданное слово.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0

**call** print

**add** **sp**, 2

**push** offset pass

**push** 200

**push** offset buffer1

**call** input

**add** **sp**, 6

**push** offset msg1

**call** print

**add** **sp**, 2

**push** offset is\_not\_space

**push** 200

**push** offset buffer2

**call** input

**add** **sp**, 6

**push** offset msg1

**call** print

**add** **sp**, 2

**push** offset is\_not\_space

**push** 200

**push** offset buffer4

**call** input

**add** **sp**, 6

**push** offset buffer3

**push** offset buffer4

**push** offset buffer2

**push** offset buffer1

**call** insert\_word\_before\_word

**add** **sp**, 6

**cmp** **ax**, -1

**jne** .ll\_1

**push** offset error\_message1

**call** print

**add** **sp**, 2

**push** offset buffer2

**call** print

**add** **sp**, 2

**push** offset error\_message2

**call** print

**add** **sp**, 2

**jmp** exit

.ll\_1:

**push** offset task\_msg1

**call** print

**add** **sp**, 2

**push** offset buffer2

**call** print

**add** **sp**, 2

**push** offset task\_msg2

**call** print

**add** **sp**, 2

**push** offset buffer4

**call** print

**add** **sp**, 2

**push** offset task\_msg3

**call** print

**add** **sp**, 2

**push** offset buffer3

**call** print

**add** **sp**, 2

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

main endp

is\_not\_space proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, **[**a1**]**

**cmp** **al**, 32

**je** e2

**pop** **bp**

**ret**

e2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_not\_space endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

insert\_word\_before\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

a4 **equ** **bp**+10

**sub** **sp**, 10

buf26 **equ** **bp**-2

buf27 **equ** **bp**-4

buf28 **equ** **bp**-6

buf29 **equ** **bp**-8

buf30 **equ** **bp**-10

**push** **[**a2**]**

**push** **[**a1**]**

**call** find

**add** **sp**, 4

**cmp** **ax**, -1

**je** \_L23

**mov** **[**buf26**]**, **ax**

**add** **[**buf26**]**, **bx**

**push** **[**a3**]**

**call** strlen

**add** **sp**, 2

**mov** **[**buf27**]**, **ax**

**mov** **ax**, **[**a4**]**

**mov** **[**buf28**]**, **ax**

**mov** **[**buf29**]**, **ax**

**mov** **ax**, **[**a1**]**

**mov** **[**buf30**]**, **ax**

**push** **[**buf26**]**

**push** **[**buf29**]**

**push** **[**buf30**]**

**call** copyA

**add** **sp**, 6

**mov** **ax**, **[**buf26**]**

**add** **[**buf30**]**, **ax**

**add** **[**buf29**]**, **ax**

**mov** **di**, **[**buf29**]**

**mov** **[di]**, **byte** ptr 32

**inc** **[**buf29**]**

**push** **[**buf29**]**

**push** **[**a3**]**

**call** copy

**add** **sp**, 4

**mov** **ax**, **[**buf27**]**

**add** **[**buf29**]**, **ax**

**push** **[**buf29**]**

**push** **[**buf30**]**

**call** copy

**add** **sp**, 4

**mov** **ax**, **[**buf28**]**

\_L23:

**add** **sp**, 10

**pop** **bp**

**ret**

insert\_word\_before\_word endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

copy proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**a2**]**

.l9:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**cmp** **dh**, 0

**jne** .l9

**pop** **bp**

**ret**

copy endp

find proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

buf16 **equ** **bp**-2

*; функция ищет в строке а1 подстроку а2*

*; и возвращает индекс подстроки*

**mov** **ax**, **[**a1**]**

**mov** **[**buf16**]**, **ax**

**dec** **[**buf16**]**

\_L17:

**inc** **[**buf16**]**

**mov** **ax**, **[**buf16**]**

**mov** **bx**, **[**a2**]**

\_L19:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **dl**, **[di]**

**inc** **ax**

**inc** **bx**

**cmp** **dl**, 0

**je** \_L20

**cmp** **dh**, 0

**je** \_L18

**cmp** **dh**, **dl**

**je** \_L19

**jne** \_L17

\_L20:

**mov** **bx**, **ax**

**sub** **bx**, **[**buf16**]**

**dec** **bx**

**mov** **ax**, **[**buf16**]**

**sub** **ax**, **[**a1**]**

**add** **sp**, 2

**pop** **bp**

**ret**

\_L18:

**mov** **ax**, -1

**add** **sp**, 2

**pop** **bp**

**ret**

find endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16 *;*

**cmp** **dh**, 0 *;*

**je** .l17 *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

buffer4 **db** 201 dup**(**0**)**

\_\_buf0 **dw** 201 dup **(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

msg1 **db** 'Enter a word: ', 10, 0

task\_msg1 **db** 'Replace the word "', 0

task\_msg2 **db** '" with the word "', 0

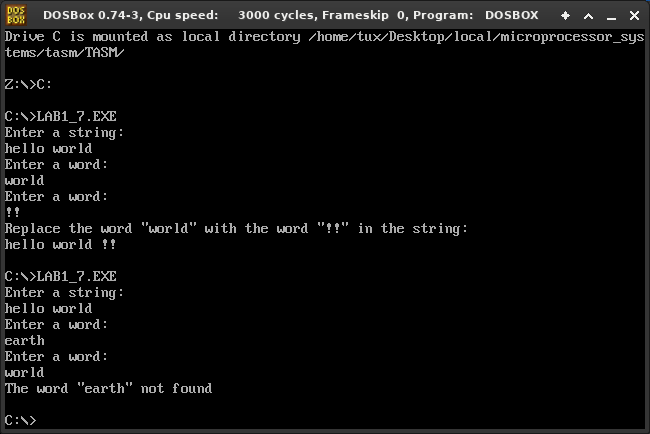
task\_msg3 **db** '" in the string: ', 10, 0

error\_message1 **db** 'The word "', 0

error\_message2 **db** '" not found', 10, 0

end main

Пример работы программы



8) Удалить во всех строках текста слова, содержащие заданный набор букв.

Код программы:

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0 *;*

**call** print *; вывод сообщения msg0*

**add** **sp**, 2 *;*

**push** offset pass *;*

**push** 200 *;*

**push** offset buffer1 *; ввод строки в буфер buffer1*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset msg1 *;*

**call** print *; вывод сообщения msg1*

**add** **sp**, 2 *;*

**push** offset is\_not\_space *;*

**push** 200 *;*

**push** offset buffer2 *; ввод строки без пробелов в буфер buffer2*

**call** input *; (buffer2 - набор букв)*

**add** **sp**, 6 *;*

**push** offset buffer3 *;*

**push** offset buffer2 *; отсортировать слова из buffer1,*

**push** offset buffer1 *; содержащие набор букв buffer2*

**call** filter *; и сохранить отсортированный массив*

**add** **sp**, 6 *; в buffer 3*

**push** offset buffer4 *;*

**push** offset buffer3 *; удалить все слова из строки,*

**push** offset buffer1 *; содержащиеся в массиве,*

**call** delete\_words *; и сохранить результат в buffer4*

**add** **sp**, 6 *;*

**push** offset task\_msg1 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer2 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset task\_msg2 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer4 *;*

**call** print *;*

**add** **sp**, 2 *;*

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

is\_not\_space proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, **[**a1**]**

**cmp** **al**, 32

**je** e2

**pop** **bp**

**ret**

e2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_not\_space endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

delete\_words proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

buf35 **equ** **bp**-2

*;*

*; функция удаляет из строки все слова из массива а2*

*; и записывает результат в a3*

*; а1 - указатель на строку*

*; а2 - указатель на массив слов в строке*

*; а3 - новая строка*

*;*

**mov** **di**, **[**a2**]**

**mov** **ax**, **[di]**

**mov** **[**buf35**]**, **ax** *; buf35 - длина массива*

**add** **[**a2**]**, **word** ptr 2

\_L31:

**cmp** **[**buf35**]**, **word** ptr 0 *;*

**je** \_L32 *; цикл (повторить buf35 раз)*

**dec** **word** ptr **[**buf35**]** *;*

**mov** **di**, **[**a2**]** *;*

**mov** **ax**, **[di]** *; ax - индекс слова в строке*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **di**, **[**a2**]** *;*

**mov** **bx**, **[di]** *; bx - длина слова*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **cx**, **[**a1**]** *; cx = a1*

**mov** **[**a1**]**, **ax** *;*

**add** **[**a1**]**, **bx** *; a1 = ax+bx*

**mov** **dx**, **[**a1**]** *;*

**sub** **dx**, **cx** *; длина части строки от сx до индекса слова в строке*

**sub** **dx**, **bx** *;*

**mov** **bx**, **[**a3**]**

**add** **[**a3**]**, **dx**

**push** **dx**

**push** **bx**

**push** **cx**

**call** copyA

**add** **sp**, 6

**jmp** \_L31 *; конец цикла*

\_L32:

**push** **[**a3**]**

**push** **[**a1**]**

**call** copy

**add** **sp**, 4

**add** **sp**, 2

**pop** **bp**

**ret**

delete\_words endp

filter proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 10

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

buf1 **equ** **bp**-2

buf2 **equ** **bp**-4

buf3 **equ** **bp**-6

buf4 **equ** **bp**-8

buf5 **equ** **bp**-10

*; входные данные:*

*; a1 - строка*

*; а2 - строка с заданным набором букв*

*; а3 - буффер массива, в котором будут слова,*

*; содержащие набор букв а2*

**push** **[**a2**]**

**call** strlen

**add** **sp**, 2

**mov** **[**buf3**]**, **ax**

**mov** **[**buf2**]**, **word** ptr 0

**mov** **ax**, **[**a3**]**

**add** **ax**, 2

**mov** **[**buf1**]**, **ax**

J1:

**push** **[**a1**]**

**call** find\_any\_word

**add** **sp**, 2

**cmp** **bx**, 0

**je** J2

**mov** **[**a1**]**, **ax**

**add** **[**a1**]**, **bx**

**mov** **[**buf4**]**, **ax**

**mov** **[**buf5**]**, **bx**

**push** **[**buf3**]**

**push** **[**a2**]**

**push** **bx**

**push** **ax**

**call** findA

**add** **sp**, 8

**cmp** **ax**, -1

**je** J1

**mov** **ax**, **[**buf4**]**

**mov** **bx**, **[**buf5**]**

**mov** **di**, **[**buf1**]***;*

**mov** **[di]**, **ax**

**add** **di**, 2

**mov** **[di]**, **bx**

**inc** **word** ptr **[**buf2**]**

**add** **[**buf1**]**, **word** ptr 4

**jmp** J1

J2:

**mov** **di**, **[**a3**]**

**mov** **ax**, **[**buf2**]**

**mov** **[di]**, **ax**

**add** **sp**, 10

**pop** **bp**

**ret**

filter endp

findA proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

a4 **equ** **bp**+10

buf16 **equ** **bp**-2

*; функция ищет в строке а1 подстроку а3*

*; и возвращает индекс подстроки*

*; a2 - размер строки а1*

*; a4 - размер строки а3*

*; если подстрока не найдена, возвращает ах=-1*

**mov** **ax**, **[**a1**]** *;*

**add** **[**a2**]**, **ax** *; a2 указывает на конец строки а1*

**mov** **ax**, **[**a3**]** *;*

**add** **[**a4**]**, **ax** *; a4 указывает на конец строки а3*

**mov** **ax**, **[**a1**]** *;*

**mov** **[**buf16**]**, **ax** *;*

**dec** **[**buf16**]** *;*

\_L17\_:

**inc** **[**buf16**]**

**mov** **ax**, **[**buf16**]**

**mov** **bx**, **[**a3**]**

\_L19\_:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **dl**, **[di]**

**cmp** **[**a4**]**, **bx**

**je** \_L20\_

**cmp** **[**a2**]**, **ax**

**je** \_L18\_

**inc** **ax**

**inc** **bx**

**cmp** **dh**, **dl**

**je** \_L19\_

**jne** \_L17\_

\_L20\_:

**mov** **bx**, **ax**

**sub** **bx**, **[**buf16**]**

**dec** **bx**

**mov** **ax**, **[**buf16**]**

**sub** **ax**, **[**a1**]**

**add** **sp**, 2

**pop** **bp**

**ret**

\_L18\_:

**mov** **ax**, -1

**add** **sp**, 2

**pop** **bp**

**ret**

findA endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

copy proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**a2**]**

.l9:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**cmp** **dh**, 0

**jne** .l9

**pop** **bp**

**ret**

copy endp

find proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

buf16 **equ** **bp**-2

*; функция ищет в строке а1 подстроку а2*

*; и возвращает индекс подстроки*

**mov** **ax**, **[**a1**]**

**mov** **[**buf16**]**, **ax**

**dec** **[**buf16**]**

\_L17:

**inc** **[**buf16**]**

**mov** **ax**, **[**buf16**]**

**mov** **bx**, **[**a2**]**

\_L19:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **dl**, **[di]**

**inc** **ax**

**inc** **bx**

**cmp** **dl**, 0

**je** \_L20

**cmp** **dh**, 0

**je** \_L18

**cmp** **dh**, **dl**

**je** \_L19

**jne** \_L17

\_L20:

**mov** **bx**, **ax**

**sub** **bx**, **[**buf16**]**

**dec** **bx**

**mov** **ax**, **[**buf16**]**

**sub** **ax**, **[**a1**]**

**add** **sp**, 2

**pop** **bp**

**ret**

\_L18:

**mov** **ax**, -1

**add** **sp**, 2

**pop** **bp**

**ret**

find endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16 *;*

**cmp** **dh**, 0 *;*

**je** .l17 *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

printA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **cx**, **[**a2**]**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

printA endp

print\_array proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

buf\_ **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **di**, **ax**

**mov** **ax**, **[di]**

**mov** **[**buf\_**]**, **ax**

**add** **[**a1**]**, 2

.l2:

**cmp** **[**buf\_**]**, 0

**je** .l3

**mov** **bx**, **[**a1**]**

**mov** **di**, **bx**

**mov** **bx**, **[di]**

**add** **[**a1**]**, 2

**mov** **cx**, **[**a1**]**

**mov** **di**, **cx**

**mov** **cx**, **[di]**

**add** **[**a1**]**, 2

**dec** **[**buf\_**]**

**push** **cx**

**push** **bx**

**call** printA

**add** **sp**, 4

**call** newline

**jmp** .l2

.l3:

**add** **sp**, 2

**pop** **bp**

**ret**

print\_array endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

buffer4 **db** 201 dup**(**0**)**

buf0 **dw** 201 dup **(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

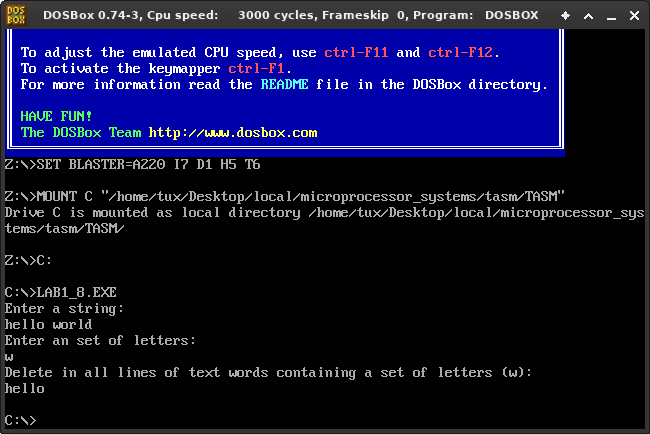
msg1 **db** 'Enter an set of letters: ', 10, 0

task\_msg1 **db** 'Delete in all lines of text words containing a set of letters (', 0

task\_msg2 **db** '): ', 10, 0

end main

Пример работы программы



9) Заменить во всех строках текста слова, содержащие заданный набор букв, на другое заданное слово.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0 *;*

**call** print *; вывод сообщения msg0*

**add** **sp**, 2 *;*

**push** offset pass *;*

**push** 200 *;*

**push** offset buffer1 *; ввод строки в буфер buffer1*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset msg1 *;*

**call** print *; вывод сообщения msg1*

**add** **sp**, 2 *;*

**push** offset is\_not\_space *;*

**push** 200 *;*

**push** offset buffer2 *; ввод строки без пробелов в буфер buffer2*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset msg2 *;*

**call** print *; вывод сообщения msg2*

**add** **sp**, 2 *;*

**push** offset is\_not\_space *;*

**push** 200 *;*

**push** offset buffer5 *; ввод строки без пробелов в буфер buffer5*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset buffer3 *;*

**push** offset buffer2 *; отсортировать слова из buffer1,*

**push** offset buffer1 *; содержащие набор букв buffer2*

**call** filter *; и сохранить отсортированный массив*

**add** **sp**, 6 *; в buffer 3*

**push** offset buffer4 *;*

**push** offset buffer5 *;*

**push** offset buffer3 *; заменить каждое слово из строки,*

**push** offset buffer1 *; содержащиеся в массиве, словом buffer5,*

**call** replace\_words *; и сохранить результат в buffer4*

**add** **sp**, 6 *;*

exit3:

**push** offset task\_msg1 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer2 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset task\_msg2 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer5 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset task\_msg3 *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer4 *;*

**call** print *;*

**add** **sp**, 2 *;*

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

is\_not\_space proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, **[**a1**]**

**cmp** **al**, 32

**je** e2

**pop** **bp**

**ret**

e2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_not\_space endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

replace\_words proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 6

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

a4 **equ** **bp**+10

buf35 **equ** **bp**-2

buf36 **equ** **bp**-4

buf37 **equ** **bp**-6

*;*

*; функция заменяет слова из массива а2 на слово а3 в строке*

*; а1 - строка*

*; а2 - массив слов в строке*

*; а3 - слово*

*; а4 - новая строка*

**push** **[**a3**]**

**call** strlen

**add** **sp**, 2

**mov** **[**buf36**]**, **ax**

**mov** **di**, **[**a2**]**

**mov** **ax**, **[di]**

**mov** **[**buf35**]**, **ax** *; buf35 - длина массива*

**add** **[**a2**]**, **word** ptr 2

\_L31:

**cmp** **[**buf35**]**, **word** ptr 0 *;*

**je** \_L32 *; цикл (повторить buf35 раз)*

**dec** **word** ptr **[**buf35**]** *;*

**mov** **di**, **[**a2**]** *;*

**mov** **ax**, **[di]** *; ax - индекс слова в строке*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **di**, **[**a2**]** *;*

**mov** **bx**, **[di]** *; bx - длина слова*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **cx**, **[**a1**]** *; cx = a1*

**mov** **[**a1**]**, **ax** *;*

**add** **[**a1**]**, **bx** *; a1 = ax+bx*

**mov** **dx**, **[**a1**]** *;*

**sub** **dx**, **cx** *; dx - длина части строки от сx до индекса слова в строке*

**sub** **dx**, **bx** *;*

**mov** **bx**, **[**a4**]**

**add** **[**a4**]**, **dx**

**push** **dx** *;*

**push** **bx** *;*

**push** **cx** *; cкопировать в новую строку от*

**call** copyA *; сx до индекса слова в строке*

**add** **sp**, 6 *;*

**push** **[**a4**]** *;*

**push** **[**a3**]** *; скопировать новое слово в*

**call** copy *; новую строку*

**add** **sp**, 4 *;*

**mov** **ax**, **[**buf36**]**

**add** **[**a4**]**, **ax**

**jmp** \_L31 *; конец цикла*

\_L32:

**push** **[**a4**]**

**push** **[**a1**]**

**call** copy

**add** **sp**, 4

**add** **sp**, 6

**pop** **bp**

**ret**

replace\_words endp

filter proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 10

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

buf1 **equ** **bp**-2

buf2 **equ** **bp**-4

buf3 **equ** **bp**-6

buf4 **equ** **bp**-8

buf5 **equ** **bp**-10

*; входные данные:*

*; a1 - строка*

*; а2 - строка с заданным набором букв*

*; а3 - буффер массива, в котором будут слова,*

*; содержащие набор букв а2*

**push** **[**a2**]**

**call** strlen

**add** **sp**, 2

**mov** **[**buf3**]**, **ax**

**mov** **[**buf2**]**, **word** ptr 0

**mov** **ax**, **[**a3**]**

**add** **ax**, 2

**mov** **[**buf1**]**, **ax**

J1:

**push** **[**a1**]**

**call** find\_any\_word

**add** **sp**, 2

**cmp** **bx**, 0

**je** J2

**mov** **[**a1**]**, **ax**

**add** **[**a1**]**, **bx**

**mov** **[**buf4**]**, **ax**

**mov** **[**buf5**]**, **bx**

**push** **[**buf3**]**

**push** **[**a2**]**

**push** **bx**

**push** **ax**

**call** findA

**add** **sp**, 8

**cmp** **ax**, -1

**je** J1

**mov** **ax**, **[**buf4**]**

**mov** **bx**, **[**buf5**]**

**mov** **di**, **[**buf1**]***;*

**mov** **[di]**, **ax**

**add** **di**, 2

**mov** **[di]**, **bx**

**inc** **word** ptr **[**buf2**]**

**add** **[**buf1**]**, **word** ptr 4

**jmp** J1

J2:

**mov** **di**, **[**a3**]**

**mov** **ax**, **[**buf2**]**

**mov** **[di]**, **ax**

**add** **sp**, 10

**pop** **bp**

**ret**

filter endp

findA proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

a4 **equ** **bp**+10

buf16 **equ** **bp**-2

*; функция ищет в строке а1 подстроку а3*

*; и возвращает индекс подстроки*

*; a2 - размер строки а1*

*; a4 - размер строки а3*

*; если подстрока не найдена, возвращает ах=-1*

**mov** **ax**, **[**a1**]** *;*

**add** **[**a2**]**, **ax** *; a2 указывает на конец строки а1*

**mov** **ax**, **[**a3**]** *;*

**add** **[**a4**]**, **ax** *; a4 указывает на конец строки а3*

**mov** **ax**, **[**a1**]** *;*

**mov** **[**buf16**]**, **ax** *;*

**dec** **[**buf16**]** *;*

\_L17\_:

**inc** **[**buf16**]**

**mov** **ax**, **[**buf16**]**

**mov** **bx**, **[**a3**]**

\_L19\_:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **dl**, **[di]**

**cmp** **[**a4**]**, **bx**

**je** \_L20\_

**cmp** **[**a2**]**, **ax**

**je** \_L18\_

**inc** **ax**

**inc** **bx**

**cmp** **dh**, **dl**

**je** \_L19\_

**jne** \_L17\_

\_L20\_:

**mov** **bx**, **ax**

**sub** **bx**, **[**buf16**]**

**dec** **bx**

**mov** **ax**, **[**buf16**]**

**sub** **ax**, **[**a1**]**

**add** **sp**, 2

**pop** **bp**

**ret**

\_L18\_:

**mov** **ax**, -1

**add** **sp**, 2

**pop** **bp**

**ret**

findA endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

copy proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**a2**]**

.l9:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**cmp** **dh**, 0

**jne** .l9

**pop** **bp**

**ret**

copy endp

find proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

buf16 **equ** **bp**-2

*; функция ищет в строке а1 подстроку а2*

*; и возвращает индекс подстроки*

**mov** **ax**, **[**a1**]**

**mov** **[**buf16**]**, **ax**

**dec** **[**buf16**]**

\_L17:

**inc** **[**buf16**]**

**mov** **ax**, **[**buf16**]**

**mov** **bx**, **[**a2**]**

\_L19:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **dl**, **[di]**

**inc** **ax**

**inc** **bx**

**cmp** **dl**, 0

**je** \_L20

**cmp** **dh**, 0

**je** \_L18

**cmp** **dh**, **dl**

**je** \_L19

**jne** \_L17

\_L20:

**mov** **bx**, **ax**

**sub** **bx**, **[**buf16**]**

**dec** **bx**

**mov** **ax**, **[**buf16**]**

**sub** **ax**, **[**a1**]**

**add** **sp**, 2

**pop** **bp**

**ret**

\_L18:

**mov** **ax**, -1

**add** **sp**, 2

**pop** **bp**

**ret**

find endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16 *;*

**cmp** **dh**, 0 *;*

**je** .l17 *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

printA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **cx**, **[**a2**]**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

printA endp

print\_array proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

buf\_ **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **di**, **ax**

**mov** **ax**, **[di]**

**mov** **[**buf\_**]**, **ax**

**add** **[**a1**]**, 2

.l2:

**cmp** **[**buf\_**]**, 0

**je** .l3

**mov** **bx**, **[**a1**]**

**mov** **di**, **bx**

**mov** **bx**, **[di]**

**add** **[**a1**]**, 2

**mov** **cx**, **[**a1**]**

**mov** **di**, **cx**

**mov** **cx**, **[di]**

**add** **[**a1**]**, 2

**dec** **[**buf\_**]**

**push** **cx**

**push** **bx**

**call** printA

**add** **sp**, 4

**call** newline

**jmp** .l2

.l3:

**add** **sp**, 2

**pop** **bp**

**ret**

print\_array endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

buffer4 **db** 201 dup**(**0**)**

buffer5 **db** 201 dup**(**0**)**

\_\_buf0 **dw** 201 dup **(**0**)**

msg0 **db** 'Enter a string: ', 10, 0

msg1 **db** 'Enter an set of letters: ', 10, 0

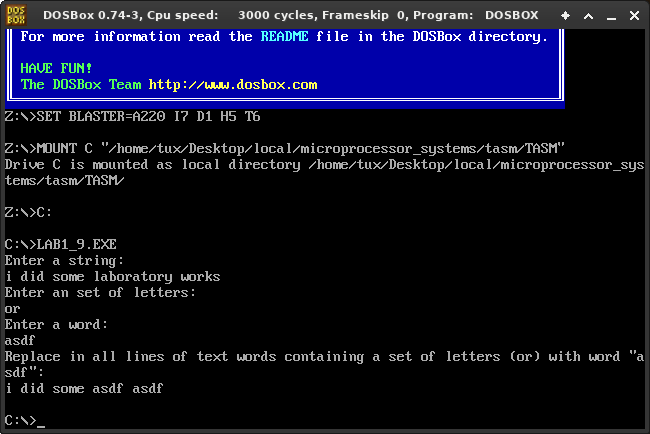
msg2 **db** 'Enter a word: ', 10, 0

task\_msg1 **db** 'Replace in all lines of text words containing a set of letters (', 0

task\_msg2 **db** ') with word "', 0

task\_msg3 **db** '": ', 10, 0

end main

 Пример работы программы

10) Вставить во всех строках текста слово "number" перед словами, состоящими из цифр.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0 *;*

**call** print *; вывод сообщения msg0*

**add** **sp**, 2 *;*

**push** offset pass *;*

**push** 200 *;*

**push** offset buffer1 *; ввод строки в буфер buffer1*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset buffer3 *; отсортировать слова из buffer1,*

**push** offset buffer1 *; содержащие только цифры*

**call** filter\_numbers *; и сохранить отсортированный массив*

**add** **sp**, 6 *; в buffer 3*

*; push offset buffer3*

*; call print\_array*

*; add sp, 2*

*; jmp exit*

**push** offset buffer4 *;*

**push** offset message *;*

**push** offset buffer3 *; вставить перед всеми числами из строки,*

**push** offset buffer1 *; содержащиеся в массиве, слово "number",*

**call** insert\_word\_after\_words *; и сохранить результат в buffer4*

**add** **sp**, 6 *;*

**push** offset task\_msg *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer4 *;*

**call** print *;*

**add** **sp**, 2 *;*

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

is\_not\_space proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, **[**a1**]**

**cmp** **al**, 32

**je** e2

**pop** **bp**

**ret**

e2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_not\_space endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

insert\_word\_after\_words proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 6

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

a4 **equ** **bp**+10

buf35 **equ** **bp**-2

buf36 **equ** **bp**-4

buf37 **equ** **bp**-6

*;*

*; функция вставляет перед всеми словами строк из массива а2 слово а3*

*; а1 - строка*

*; а2 - массив слов в строке*

*; а3 - слово*

*; а4 - новая строка*

**push** **[**a3**]**

**call** strlen

**add** **sp**, 2

**mov** **[**buf36**]**, **ax**

**mov** **di**, **[**a2**]**

**mov** **ax**, **[di]**

**mov** **[**buf35**]**, **ax** *; buf35 - длина массива*

**add** **[**a2**]**, **word** ptr 2

\_L31:

**cmp** **[**buf35**]**, **word** ptr 0 *;*

**je** \_L32 *; цикл (повторить buf35 раз)*

**dec** **word** ptr **[**buf35**]** *;*

**mov** **di**, **[**a2**]** *;*

**mov** **ax**, **[di]** *; ax - индекс слова в строке*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **di**, **[**a2**]** *;*

**mov** **bx**, **[di]** *; bx - длина слова*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **cx**, **[**a1**]** *; cx = a1*

**mov** **[**a1**]**, **ax** *;*

**add** **[**a1**]**, **bx** *; a1 = ax+bx*

**mov** **dx**, **[**a1**]** *;*

**add** **dx**, **bx** *;*

**sub** **dx**, **cx** *; dx - длина части строки от сx до индекса конца слова в строке*

**sub** **dx**, **bx** *;*

**mov** **bx**, **[**a4**]**

**add** **[**a4**]**, **dx**

**push** **dx** *;*

**push** **bx** *;*

**push** **cx** *; cкопировать в новую строку от*

**call** copyA *; сx до индекса конца слова в строке*

**add** **sp**, 6 *;*

**push** **[**a4**]** *;*

**push** **[**a3**]** *; скопировать новое слово в*

**call** copy *; новую строку*

**add** **sp**, 4 *;*

**mov** **ax**, **[**buf36**]**

**add** **[**a4**]**, **ax**

**jmp** \_L31 *; конец цикла*

\_L32:

**push** **[**a4**]**

**push** **[**a1**]**

**call** copy

**add** **sp**, 4

**add** **sp**, 6

**pop** **bp**

**ret**

insert\_word\_after\_words endp

filter\_numbers proc

*; входные данные:*

*; a1 - строка*

*; а2 - буффер массива, в котором будут слова,*

*; содержащие только цифры*

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**sub** **sp**, 4

buf1 **equ** **bp**-2

buf2 **equ** **bp**-4

**mov** **[**buf2**]**, 0

**mov** **ax**, **[**a2**]**

**add** **ax**, 2

**mov** **[**buf1**]**, **ax**

J1:

**push** **[**a1**]**

**call** find\_any\_word

**add** **sp**, 2

**cmp** **bx**, 0

**je** J2

**push** **bx**

**push** **ax**

**call** is\_num

**mov** **cx**, **ax**

**pop** **ax**

**pop** **bx**

**mov** **[**a1**]**, **ax**

**add** **[**a1**]**, **bx**

**cmp** **cx**, 0

**je** J1

**mov** **di**, **[**buf1**]***;*

**mov** **[di]**, **ax**

**add** **di**, 2

**mov** **[di]**, **bx**

**inc** **word** ptr **[**buf2**]**

**add** **[**buf1**]**, 4

**jmp** J1

J2:

**mov** **di**, **[**a2**]**

**mov** **ax**, **[**buf2**]**

**mov** **[di]**, **ax**

**add** **sp**, 4

**pop** **bp**

**ret**

filter\_numbers endp

is\_num proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

*; проверяет, является ли слово номером*

*; возвращает 1 если да*

*; 0 если нет*

**mov** **bx**, **[**a2**]**

**mov** **di**, **[**a1**]**

\_\_l:

**cmp** **bx**, 0

**je** .\_U1

**mov** **al**, **[di]**

**cmp** **al**, '0'

**jl** .\_U2

**cmp** **al**, '9'

**jg** .\_U2

**dec** **bx**

**inc** **di**

**jmp** \_\_l

.\_U1:

**mov** **ax**, 1

**pop** **bp**

**ret**

.\_U2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_num endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

copy proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**a2**]**

.l9:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**cmp** **dh**, 0

**jne** .l9

**pop** **bp**

**ret**

copy endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16l: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16l *;*

**cmp** **dh**, 0 *;*

**je** .l17l *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18l:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18l

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17l:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

buffer4 **db** 201 dup**(**0**)**

buffer5 **db** 201 dup**(**0**)**

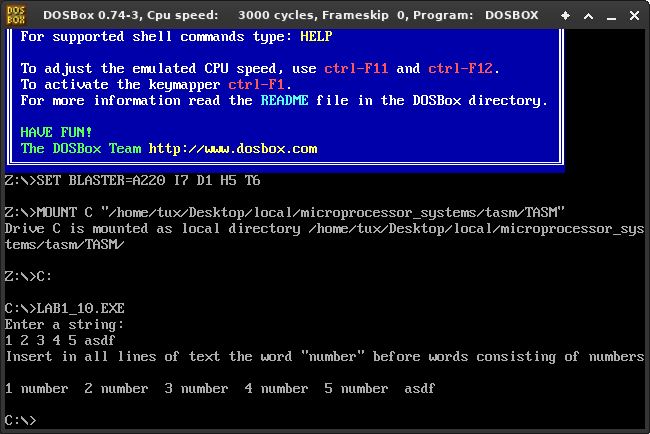
message **db** ' number ', 0

msg0 **db** 'Enter a string: ', 10, 0

task\_msg **db** 'Insert in all lines of text the word "number" before words consisting of numbers',10, 0

end main

Пример работы программы



11) Удалить во всех строках текста слова являющиеся числами.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0 *;*

**call** print *; вывод сообщения msg0*

**add** **sp**, 2 *;*

**push** offset pass *;*

**push** 200 *;*

**push** offset buffer1 *; ввод строки в буфер buffer1*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset buffer3 *; отсортировать слова из buffer1,*

**push** offset buffer1 *; содержащие только цифры*

**call** filter\_numbers *; и сохранить отсортированный массив*

**add** **sp**, 6 *; в buffer 3*

*; push offset buffer3*

*; call print\_array*

*; add sp, 2*

*; jmp exit*

**push** offset buffer4 *;*

**push** offset buffer3 *; удалить все числа из строки,*

**push** offset buffer1 *; содержащиеся в массиве, слово "number",*

**call** delete\_words *; и сохранить результат в buffer4*

**add** **sp**, 6 *;*

**push** offset task\_msg *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer4 *;*

**call** print *;*

**add** **sp**, 2 *;*

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

is\_not\_space proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, **[**a1**]**

**cmp** **al**, 32

**je** e2

**pop** **bp**

**ret**

e2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_not\_space endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

delete\_words proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

buf35 **equ** **bp**-2

*;*

*; функция удаляет из строки все слова из массива а2*

*; и записывает результат в a3*

*; а1 - указатель на строку*

*; а2 - указатель на массив слов в строке*

*; а3 - новая строка*

*;*

**mov** **di**, **[**a2**]**

**mov** **ax**, **[di]**

**mov** **[**buf35**]**, **ax** *; buf35 - длина массива*

**add** **[**a2**]**, **word** ptr 2

\_L31:

**cmp** **[**buf35**]**, **word** ptr 0 *;*

**je** \_L32 *; цикл (повторить buf35 раз)*

**dec** **word** ptr **[**buf35**]** *;*

**mov** **di**, **[**a2**]** *;*

**mov** **ax**, **[di]** *; ax - индекс слова в строке*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **di**, **[**a2**]** *;*

**mov** **bx**, **[di]** *; bx - длина слова*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **cx**, **[**a1**]** *; cx = a1*

**mov** **[**a1**]**, **ax** *;*

**add** **[**a1**]**, **bx** *; a1 = ax+bx*

**mov** **dx**, **[**a1**]** *;*

**sub** **dx**, **cx** *; длина части строки от сx до индекса слова в строке*

**sub** **dx**, **bx** *;*

**mov** **bx**, **[**a3**]**

**add** **[**a3**]**, **dx**

**push** **dx**

**push** **bx**

**push** **cx**

**call** copyA

**add** **sp**, 6

**jmp** \_L31 *; конец цикла*

\_L32:

**push** **[**a3**]**

**push** **[**a1**]**

**call** copy

**add** **sp**, 4

**add** **sp**, 2

**pop** **bp**

**ret**

delete\_words endp

filter\_numbers proc

*; входные данные:*

*; a1 - строка*

*; а2 - буффер массива, в котором будут слова,*

*; содержащие только цифры*

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**sub** **sp**, 4

buf1 **equ** **bp**-2

buf2 **equ** **bp**-4

**mov** **[**buf2**]**, 0

**mov** **ax**, **[**a2**]**

**add** **ax**, 2

**mov** **[**buf1**]**, **ax**

J1:

**push** **[**a1**]**

**call** find\_any\_word

**add** **sp**, 2

**cmp** **bx**, 0

**je** J2

**push** **bx**

**push** **ax**

**call** is\_num

**mov** **cx**, **ax**

**pop** **ax**

**pop** **bx**

**mov** **[**a1**]**, **ax**

**add** **[**a1**]**, **bx**

**cmp** **cx**, 0

**je** J1

**mov** **di**, **[**buf1**]***;*

**mov** **[di]**, **ax**

**add** **di**, 2

**mov** **[di]**, **bx**

**inc** **word** ptr **[**buf2**]**

**add** **[**buf1**]**, 4

**jmp** J1

J2:

**mov** **di**, **[**a2**]**

**mov** **ax**, **[**buf2**]**

**mov** **[di]**, **ax**

**add** **sp**, 4

**pop** **bp**

**ret**

filter\_numbers endp

is\_num proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

*; проверяет, является ли слово номером*

*; возвращает 1 если да*

*; 0 если нет*

**mov** **bx**, **[**a2**]**

**mov** **di**, **[**a1**]**

\_\_l:

**cmp** **bx**, 0

**je** .\_U1

**mov** **al**, **[di]**

**cmp** **al**, '0'

**jl** .\_U2

**cmp** **al**, '9'

**jg** .\_U2

**dec** **bx**

**inc** **di**

**jmp** \_\_l

.\_U1:

**mov** **ax**, 1

**pop** **bp**

**ret**

.\_U2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_num endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

copy proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**a2**]**

.l9:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**cmp** **dh**, 0

**jne** .l9

**pop** **bp**

**ret**

copy endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16l: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16l *;*

**cmp** **dh**, 0 *;*

**je** .l17l *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18l:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18l

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17l:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

buffer4 **db** 201 dup**(**0**)**

buffer5 **db** 201 dup**(**0**)**

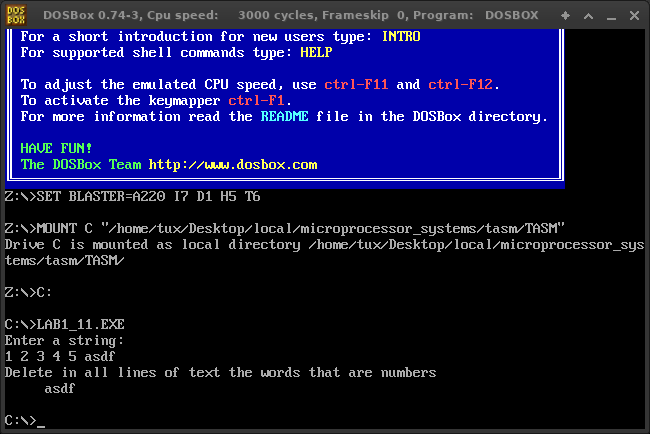
message **db** ' number ', 0

msg0 **db** 'Enter a string: ', 10, 0

task\_msg **db** 'Delete in all lines of text the words that are numbers',10, 0

end main

Пример работы программы



12 Удалить во всех соках текста слова являющиеся числами.

Код программы

.model **small**

.**stack** 100h

.**code**

main proc

**mov** **ax**, @**data**

**mov** **ds**, **ax**

**push** offset msg0 *;*

**call** print *; вывод сообщения msg0*

**add** **sp**, 2 *;*

**push** offset pass *;*

**push** 200 *;*

**push** offset buffer1 *; ввод строки в буфер buffer1*

**call** input *;*

**add** **sp**, 6 *;*

**push** offset buffer3 *; отсортировать слова из buffer1,*

**push** offset buffer1 *; содержащие только цифры*

**call** filter\_numbers *; и сохранить отсортированный массив*

**add** **sp**, 6 *; в buffer 3*

*; push offset buffer3*

*; call print\_array*

*; add sp, 2*

*; jmp exit*

**push** offset buffer4 *;*

**push** offset buffer3 *; удалить все числа из строки,*

**push** offset buffer1 *; содержащиеся в массиве, слово "number",*

**call** delete\_words *; и сохранить результат в buffer4*

**add** **sp**, 6 *;*

**push** offset task\_msg *;*

**call** print *;*

**add** **sp**, 2 *;*

**push** offset buffer4 *;*

**call** print *;*

**add** **sp**, 2 *;*

**call** newline

exit:

**mov** **ax**,4C00h

**int** 21h

**ret**

main endp

is\_not\_space proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, **[**a1**]**

**cmp** **al**, 32

**je** e2

**pop** **bp**

**ret**

e2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_not\_space endp

newline proc

**mov** **ah**, 06h *;*

**mov** **dl**, 13 *; вывести новую линию*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 10 *;*

**int** 21h *;*

newline endp

pass proc

**push** **bp**

**mov** **bp**, **sp**

**mov** **ax**, **[bp**+4**]**

**pop** **bp**

**ret**

pass endp

delete\_words proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

buf35 **equ** **bp**-2

*;*

*; функция удаляет из строки все слова из массива а2*

*; и записывает результат в a3*

*; а1 - указатель на строку*

*; а2 - указатель на массив слов в строке*

*; а3 - новая строка*

*;*

**mov** **di**, **[**a2**]**

**mov** **ax**, **[di]**

**mov** **[**buf35**]**, **ax** *; buf35 - длина массива*

**add** **[**a2**]**, **word** ptr 2

\_L31:

**cmp** **[**buf35**]**, **word** ptr 0 *;*

**je** \_L32 *; цикл (повторить buf35 раз)*

**dec** **word** ptr **[**buf35**]** *;*

**mov** **di**, **[**a2**]** *;*

**mov** **ax**, **[di]** *; ax - индекс слова в строке*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **di**, **[**a2**]** *;*

**mov** **bx**, **[di]** *; bx - длина слова*

**add** **[**a2**]**, **word** ptr 2 *;*

**mov** **cx**, **[**a1**]** *; cx = a1*

**mov** **[**a1**]**, **ax** *;*

**add** **[**a1**]**, **bx** *; a1 = ax+bx*

**mov** **dx**, **[**a1**]** *;*

**sub** **dx**, **cx** *; длина части строки от сx до индекса слова в строке*

**sub** **dx**, **bx** *;*

**mov** **bx**, **[**a3**]**

**add** **[**a3**]**, **dx**

**push** **dx**

**push** **bx**

**push** **cx**

**call** copyA

**add** **sp**, 6

**jmp** \_L31 *; конец цикла*

\_L32:

**push** **[**a3**]**

**push** **[**a1**]**

**call** copy

**add** **sp**, 4

**add** **sp**, 2

**pop** **bp**

**ret**

delete\_words endp

filter\_numbers proc

*; входные данные:*

*; a1 - строка*

*; а2 - буффер массива, в котором будут слова,*

*; содержащие только цифры*

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**sub** **sp**, 4

buf1 **equ** **bp**-2

buf2 **equ** **bp**-4

**mov** **[**buf2**]**, 0

**mov** **ax**, **[**a2**]**

**add** **ax**, 2

**mov** **[**buf1**]**, **ax**

J1:

**push** **[**a1**]**

**call** find\_any\_word

**add** **sp**, 2

**cmp** **bx**, 0

**je** J2

**push** **bx**

**push** **ax**

**call** is\_num

**mov** **cx**, **ax**

**pop** **ax**

**pop** **bx**

**mov** **[**a1**]**, **ax**

**add** **[**a1**]**, **bx**

**cmp** **cx**, 0

**je** J1

**mov** **di**, **[**buf1**]***;*

**mov** **[di]**, **ax**

**add** **di**, 2

**mov** **[di]**, **bx**

**inc** **word** ptr **[**buf2**]**

**add** **[**buf1**]**, 4

**jmp** J1

J2:

**mov** **di**, **[**a2**]**

**mov** **ax**, **[**buf2**]**

**mov** **[di]**, **ax**

**add** **sp**, 4

**pop** **bp**

**ret**

filter\_numbers endp

is\_num proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

*; проверяет, является ли слово номером*

*; возвращает 1 если да*

*; 0 если нет*

**mov** **bx**, **[**a2**]**

**mov** **di**, **[**a1**]**

\_\_l:

**cmp** **bx**, 0

**je** .\_U1

**mov** **al**, **[di]**

**cmp** **al**, '0'

**jl** .\_U2

**cmp** **al**, '9'

**jg** .\_U2

**dec** **bx**

**inc** **di**

**jmp** \_\_l

.\_U1:

**mov** **ax**, 1

**pop** **bp**

**ret**

.\_U2:

**mov** **ax**, 0

**pop** **bp**

**ret**

is\_num endp

copyA proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

a3 **equ** **bp**+8

*; функция копирует а3 символов из а1 строки в а2 строку*

**mov** **ax**, **[**a1**]***;ax - указатель на строку а1*

**mov** **bx**, **[**a2**]***;bx - указатель на строку а2*

**mov** **cx**, **[**a3**]**

**add** **cx**, **ax** *;cx - указатель на конец строки а1*

.\_l9:

**cmp** **ax**, **cx**

**je** .\_l10

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**jmp** .\_l9

.\_l10:

**pop** **bp**

**ret**

copyA endp

copy proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

a2 **equ** **bp**+6

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**a2**]**

.l9:

**mov** **di**, **ax**

**mov** **dh**, **[di]**

**mov** **di**, **bx**

**mov** **[di]**, **dh**

**inc** **ax**

**inc** **bx**

**cmp** **dh**, 0

**jne** .l9

**pop** **bp**

**ret**

copy endp

print proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**push** **[**a1**]**

**call** strlen

**add** **sp**, 2

**mov** **cx**, **ax**

**mov** **ah**, 40h

**mov** **dx**, **[**a1**]**

**mov** **bx**, 1

**int** 21h

**pop** **bp**

**ret**

print endp

strlen proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

**mov** **ax**, 0

**mov** **ax**, **[**a1**]**

.l1:

**mov** **di**, **ax**

**mov** **dl**, **[di]**

**inc** **ax**

**cmp** **dl**, 0

**jne** .l1

**sub** **ax**, **[**a1**]**

**dec** **ax**

**pop** **bp**

**ret**

strlen endp

find\_any\_word proc

**push** **bp**

**mov** **bp**, **sp**

a1 **equ** **bp**+4

*; функция ищет первое слово*

*; в строке а1 и возвращает*

*; индекс и размер слова в строке а1*

*; если слово не найдено, возвращается 0*

**mov** **ax**, **[**a1**]**

.l16l: *;*

**mov** **di**, **ax** *;*

**mov** **dh**, **[di]** *;*

**inc** **ax** *;*

**cmp** **dh**, 32 *; пропустить пробелы*

**je** .l16l *;*

**cmp** **dh**, 0 *;*

**je** .l17l *;*

**dec** **ax** *;*

**mov** **bx**, **ax** *; ax - индекс слова в строке*

.l18l:

**inc** **bx** *;*

**mov** **di**, **bx** *;*

**mov** **dh**, **[di]** *;*

**cmp** **dh**, 32 *;*

**je** .l19 *;*

**cmp** **dh**, 0 *;*

**je** .l19 *;*

**jmp** .l18l

.l19:

**sub** **bx**, **ax**

**pop** **bp**

**ret**

.l17l:

**mov** **ax**, 0

**mov** **bx**, 0

**pop** **bp**

**ret**

find\_any\_word endp

input proc

**push** **bp**

**mov** **bp**, **sp**

**sub** **sp**, 2 *; зарезервировать для новой переменной*

a1 **equ** **bp**+4 *; a1 - указатель на начало буффер*

a2 **equ** **bp**+6 *; a2 - длина буффера*

a3 **equ** **bp**+8 *; а3 - функция, которая сбрасывает недопустимые символы*

buf0 **equ** **bp**-2 *; новая переменная (указатель)*

**mov** **ax**, **[**a1**]**

**mov** **[**buf0**]**, **ax** *; сохраняем значение а1 в buf0*

**add** **[**a2**]**, **ax** *; помещаем в а2 указатель на конец буффера*

.\_ll10:

**mov** **ah**, 07h *; ввод*

**int** 21h

**cmp** **al**, 8 *; если нажата кнопка backspace, то*

**je** .\_l12 *; выполнить .\_l12*

**cmp** **al**, 13 *; если нажат enter, то*

**je** .\_l15 *; выполнить .\_l15*

**push** **ax**

**call** **[**a3**]** *; проверить символ на недопустимые значения*

**add** **sp**, 2

**cmp** **al**, 0 *; если символ недопустимый, то оно обращается в ноль*

**je** .\_ll10

**mov** **si**, **[**a1**]** *; записать символ*

**mov** **[si]**, **al** *;*

**mov** **ah**, 06h

**mov** **dl**, **al** *; вывести символ*

**int** 21h

**inc** **word** ptr **[**a1**]** *; переместить указатель вперед*

**mov** **ax**, **[**a1**]** *;*

**mov** **bx**, **[**a2**]** *;*

**cmp** **ax**, **bx** *; если указатель в конце строки, то*

**je** .\_l14 *; выполнить .\_l14*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l12:

**mov** **ax**, **[**a1**]**

**mov** **bx**, **[**buf0**]**

**cmp** **ax**, **bx** *; eсли указатель в начале строки, то*

**je** .\_ll10 *; вернуться в .\_l10*

**dec** **word** ptr **[**a1**]** *; cместить указатель назад*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 32 *; cтереть символ*

**int** 21h *;*

**mov** **ah**, 06h *;*

**mov** **dl**, 8 *;*

**int** 21h *;*

**jmp** .\_ll10 *; вернуться в .\_l10*

.\_l14:

**mov** **ah**, 07h *;*

**int** 21h *; выполняется до тех пор,*

**cmp** **al**, 8 *; пока не нажата кнопка backspace*

**je** .\_l12 *; если нажата кнопка enter,*

**cmp** **al**, 13 *; то вывести новую линию*

**je** .\_l15 *; записать 0 в конец строки и завершить*

**jmp** .\_l14

.\_l15:

**call** newline *; вывести новую линию*

**mov** **di**, **[**a1**]**

**mov** **[di]**, **byte** ptr 0 *; записать 0 в конец строки*

**add** **sp**, 2 *;*

**pop** **bp** *; завершить подпрограмму, освободить стек*

**ret** *;*

input endp

**.data**

buffer1 **db** 201 dup**(**0**)**

buffer2 **db** 201 dup**(**0**)**

buffer3 **db** 201 dup**(**0**)**

buffer4 **db** 201 dup**(**0**)**

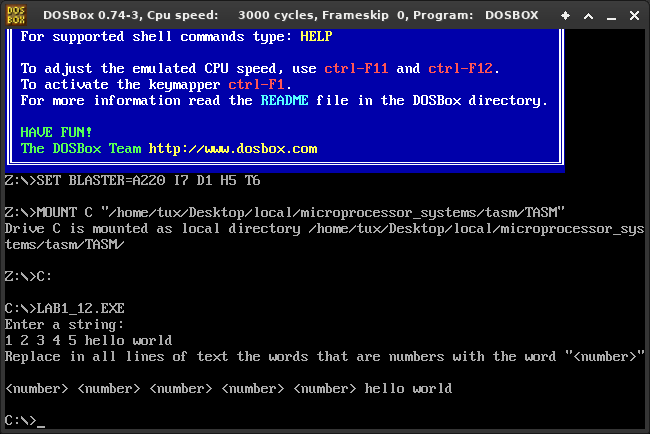
buffer5 **db** 201 dup**(**0**)**

message **db** ' number ', 0

msg0 **db** 'Enter a string: ', 10, 0

task\_msg **db** 'Delete in all lines of text the words that are numbers',10, 0

end main

Пример работы программы