# Л/р №6. Обработка символьных строк

**1.**

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

MOV AX, 00h

CALL TASK1

MOV AX, 20h

CALL TASK1

MOV AX, 40h

CALL TASK1

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

TASK1:

MOV BYTE[SI], BX

RET

;===[ Начало сегмента данных ]==========================================

**2.**

%macro unimacro 2

MOV AX, %2

CALL %1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

unimacro TASK2, 00h

unimacro TASK2, 20h

unimacro TASK2, 40h

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

TASK2:

MOV SI, AX

MOV AX, 00h

MOV [SI], AX

RET

;===[ Начало сегмента данных ]==========================================

**3**. (Формат строки C-подобный)

%macro unimacro 2

MOV AX, %2

CALL %1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

unimacro TASK3, 00h

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

TASK3:

MOV SI, AX

MOV CX, 15

NEXT1:

MOV BYTE[SI], '#'

INC SI

LOOP NEXT1

MOV BYTE[SI], 00h

RET

;===[ Начало сегмента данных ]==========================================

**4.** (Два варианта: формат Pascal-подобный и C-подобный)

%macro unimacro 2

MOV AX, %2

CALL %1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

unimacro TASK3\_PAS, 00h ; use task3 for init strings

unimacro TASK3\_C, 20h ; use task3 for init strings

unimacro TASK41, 00h

unimacro TASK42, 20h

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

TASK3\_PAS:

MOV SI, AX

MOV CX, 12

MOV [SI], CX

INC SI

NEXT1:

MOV BYTE[SI], '#'

INC SI

LOOP NEXT1

RET

TASK3\_C:

MOV SI, AX

MOV CX, 15

NEXT11:

MOV BYTE[SI], '#'

INC SI

LOOP NEXT11

MOV BYTE[SI], 00h

RET

TASK41:

MOV SI, AX

MOV BL, BYTE[SI] ; result in BL

RET

TASK42:

MOV SI, AX

MOV BX, 0

NEXT2:

MOV CL, BYTE[SI + BX]

INC BX

CMP CL, 00h

JNE NEXT2

DEC BX ; dec becouse last char is 00h

MOV CX, BX ; result in CX

RET

;===[ Начало сегмента данных ]==========================================

**5.** (Формат строки C-подобный)

%macro unimacro 3

MOV AX, %2

MOV BX, %3

CALL %1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

MOV AX, 00h

CALL INITSTR

unimacro TASK5, 00h, 'H'

unimacro TASK5, 00h, 'E'

unimacro TASK5, 00h, 'L'

unimacro TASK5, 00h, 'L'

unimacro TASK5, 00h, 'O'

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR:

MOV SI, AX

MOV BYTE[SI], 00h

RET

TASK5:

MOV SI, AX

MOV CX, BX

MOV BX, 0

NEXT1:

MOV DL, BYTE[SI+BX]

INC BX

CMP DL, 00h

JNE NEXT1

DEC BX

MOV [SI+BX], CX

MOV BYTE[SI+BX+1], 00h

RET

;===[ Начало сегмента данных ]==========================================

**6.** (Формат строки C-подобный)

%macro unimacro 3

MOV AX, %2

MOV BX, %3

CALL %1

%endmacro

%macro initstr 1

MOV AX, %1

CALL INITSTR1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

initstr 00h

unimacro TASK6, 00h, 'H'

unimacro TASK6, 00h, 'E'

unimacro TASK6, 00h, 'L'

unimacro TASK6, 00h, 'L'

unimacro TASK6, 00h, 'O'

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR1:

MOV SI, AX

MOV BX, 0

MOV CX, 16

MOV AX, 'A'

NEXTCHAR:

MOV BYTE[SI+BX], AL

INC BX

INC AX

LOOP NEXTCHAR

MOV CX, 12

NEXTCHAR2:

MOV BYTE[SI+BX], AL

INC BX

DEC AX

LOOP NEXTCHAR2

MOV BYTE[SI+BX], 00h

RET

TASK6:

MOV SI, AX

MOV CX, BX

MOV BX, 0

NEXT1:

CMP BX, 31

JE SETNULL

MOV DL, BYTE[SI+BX]

INC BX

CMP DL, 00h

JNE NEXT1

MOV [SI+BX-1], CX

SETNULL:

MOV BYTE[SI+BX], 00h

RET

;===[ Начало сегмента данных ]==========================================

**7.** (Формат строки C-подобный)

%macro unimacro 3

MOV AX, %2

MOV BX, %3

CALL %1

%endmacro

%macro initstr 1

MOV AX, %1

CALL INITSTR1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

initstr 00h

unimacro TASK7, 00h, '#'

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR1:

MOV SI, AX

MOV BX, 0

MOV CX, 16

MOV AX, 'A'

NEXTCHAR:

MOV BYTE[SI+BX], AL

INC BX

INC AX

LOOP NEXTCHAR

MOV CX, 12

NEXTCHAR2:

MOV BYTE[SI+BX], AL

INC BX

DEC AX

LOOP NEXTCHAR2

MOV BYTE[SI+BX], 00h

RET

TASK7:

MOV SI, AX

MOV DL, BYTE[SI] ; save prev

MOV BYTE[SI], BL

MOV BX, 1

NEXT1:

CMP BX, 31

JE SETNULL

MOV AL, BYTE[SI+BX] ; save actual

MOV BYTE[SI+BX], DL ; set prev

MOV DL, AL ; set prev = actual

INC BX

CMP DL, 00h

JNE NEXT1

SETNULL:

MOV BYTE[SI+BX], 00h

RET

;===[ Начало сегмента данных ]==========================================

**8.** (Формат строки C-подобный)

%macro unimacro 4

MOV AX, %2

MOV BX, %3

MOV CX, %4

CALL %1

%endmacro

%macro initstr 1

MOV AX, %1

CALL INITSTR1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

initstr 00h

unimacro TASK8, 00h, 05h, '#'

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR1:

MOV SI, AX

MOV BX, 0

MOV CX, 16

MOV AX, 'A'

NEXTCHAR:

MOV BYTE[SI+BX], AL

INC BX

INC AX

LOOP NEXTCHAR

MOV CX, 12

NEXTCHAR2:

MOV BYTE[SI+BX], AL

INC BX

DEC AX

LOOP NEXTCHAR2

MOV BYTE[SI+BX], 00h

RET

TASK8:

MOV SI, AX

CMP BX, 0

JB EXIT

CMP BX, 31

JA EXIT

MOV DL, BYTE[SI+BX-1] ; save prev

MOV BYTE[SI+BX-1], CL

NEXT1:

CMP BX, 31

JE SETNULL

MOV AL, BYTE[SI+BX] ; save actual

MOV BYTE[SI+BX], DL ; set prev

MOV DL, AL ; set prev = actual

INC BX

CMP DL, 00h

JNE NEXT1

SETNULL:

MOV BYTE[SI+BX], 00h

EXIT:

RET

;===[ Начало сегмента данных ]==========================================

**9.** (Формат строки C-подобный)

%macro unimacro 3

MOV AX, %2

MOV BX, %3

CALL %1

%endmacro

%macro initstr 1

MOV AX, %1

CALL INITSTR1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

initstr 40h

unimacro TASK9, 40h, '#'

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR1:

MOV SI, AX

MOV BX, 0

MOV CX, 16

MOV AX, 'A'

NEXTCHAR:

MOV BYTE[SI+BX], AL

INC BX

INC AX

LOOP NEXTCHAR

MOV BYTE[SI+BX], '#'

INC BX

MOV CX, 12

NEXTCHAR2:

MOV BYTE[SI+BX], AL

INC BX

DEC AX

LOOP NEXTCHAR2

MOV BYTE[SI+BX], 00h

RET

TASK9:

MOV SI, AX

MOV DL, BL

MOV BX, 0

NEXT1:

CMP BX, 31

JE EXIT

MOV AL, BYTE[SI+BX]

INC BX

CMP AL, 00h

JE EXIT

CMP AL, DL

JNE NEXT1

MOV AL, BL ; found! copy to AL

EXIT:

RET

;===[ Начало сегмента данных ]==========================================

**10.** (Формат строки C-подобный)

%macro unimacro 3

MOV AX, %2

MOV BX, %3

CALL %1

%endmacro

%macro initstr 1

MOV AX, %1

CALL INITSTR1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

initstr 40h

unimacro TASK10, 40h, '#'

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR1:

MOV SI, AX

MOV BX, 0

MOV CX, 14

MOV BYTE[SI+BX], '#'

INC BX

MOV AX, 'A'

NEXTCHAR:

MOV BYTE[SI+BX], AL

INC BX

INC AX

LOOP NEXTCHAR

MOV BYTE[SI+BX], '#'

INC BX

MOV CX, 12

NEXTCHAR2:

MOV BYTE[SI+BX], AL

INC BX

DEC AX

LOOP NEXTCHAR2

MOV BYTE[SI+BX], '#'

INC BX

MOV BYTE[SI+BX], 00h

RET

TASK10:

MOV SI, AX

MOV DL, BL

MOV BX, 0

NEXT1:

CMP BX, 31

JE EXIT

MOV AL, BYTE[SI+BX]

INC BX

CMP AL, 00h

JE EXIT

CMP AL, DL

JNE NEXT1

MOV CL, BL; found! copy to CL

JMP NEXT1

EXIT:

RET

;===[ Начало сегмента данных ]==========================================

**11.** (Формат строки C-подобный)

%macro unimacro 4

MOV AX, %2

MOV BX, %3

MOV CX, %4

CALL %1

%endmacro

%macro initstr 1

MOV AX, %1

CALL INITSTR1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

initstr 20h

unimacro TASK11, 20h, '#', 5

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR1:

MOV SI, AX

MOV BX, 0

MOV DX, 0

MOV AX, 'A'

REPEAT:

MOV BYTE[SI+BX], '#'

INC BX

MOV CX, 5

NEXTCHAR:

MOV BYTE[SI+BX], AL

INC BX

INC AX

LOOP NEXTCHAR

INC DX

CMP DX, 4

JNE REPEAT

MOV BYTE[SI+BX], '#'

INC BX

MOV BYTE[SI+BX], 00h

RET

TASK11:

MOV SI, AX

MOV DL, BL ; symbol

MOV DI, 0

MOV BX, 0

NEXT1:

CMP BX, 31

JE EXIT

MOV AL, BYTE[SI+BX]

INC BX

CMP AL, 00h

JE EXIT

CMP AL, DL

JNE NEXT1

INC DI

CMP CX, DI

JNE NEXT1

MOV AL, BL ; found! copy position to AL

EXIT:

RET

;===[ Начало сегмента данных ]==========================================

**12.** (Формат строки C-подобный)

%macro unimacro 3

MOV AX, %2

MOV BX, %3

CALL %1

%endmacro

%macro initstr 1

MOV AX, %1

CALL INITSTR1

%endmacro

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START:

;---[ Точка старта ]-------------------------------------------------------------------—

initstr 00h

unimacro TASK12, 00h, '#'

;---[ Стандартное завершение программы - НЕ СТИРАТЬ НИКОГДА !!! ]----------------------—

mov AX, 4C00h

int 21h

INITSTR1:

MOV SI, AX

MOV BX, 0

MOV DX, 0

MOV AX, 'A'

REPEAT:

MOV BYTE[SI+BX], '#'

INC BX

MOV CX, 5

NEXTCHAR:

MOV BYTE[SI+BX], AL

INC BX

INC AX

LOOP NEXTCHAR

INC DX

CMP DX, 4

JNE REPEAT

MOV BYTE[SI+BX], '#'

INC BX

MOV BYTE[SI+BX], 00h

RET

TASK12:

MOV SI, AX

MOV DL, BL ; symbol

MOV CX, 0

MOV BX, 0

NEXT1:

CMP BX, 31

JE EXIT

MOV AL, BYTE[SI+BX]

INC BX

CMP AL, 00h

JE EXIT

CMP AL, DL

JNE NEXT1

INC CX

JMP NEXT1

EXIT:

MOV AX, CX

RET

;===[ Начало сегмента данных ]==========================================