

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей  
Кафедра информатики  
Дисциплина «Конструирование программ»

## **ОТЧЕТ**

к лабораторной работе №4

на тему:

**«ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ»**

БГУИР 1-40-04-01

Выполнил студент группы 253504  
ЛЯНГО Иван Геннадьевич

---

(дата, подпись студента)

Проверил ассистент кафедры  
информатики  
РОМАНЮК Максим Валерьевич

---

(дата, подпись преподавателя)

Минск 2023

**Цель работы:** Задание 1. Вариант 1. Отсортировать символы в строке по значению ASCII кода символа.

**Ход работы:** на рисунке 1 представлены изначальные значения регистров, на рисунке 2 – изначальные значения переменных, на рисунках 3 и 4 – соответствующие значения после выполнения программы, на рисунке 5 – окно для работы с программой.

Листинг 1 – Исходный код программы задания 1

```
.model tiny
.code
org 100h
start:
    call read_length
    mov length,bx
    cmp length,200
    ja error
    mov ah,9
    lea dx,new_line
    int 21h
    mov di,0
read_string:
    cmp di,length
    je after_read_string
    mov ah,08h
    int 21h
    mov [buff+di],al
    mov ah,02h
    mov dl,[buff+di]
    int 21h
    inc di
    jmp read_string
after_read_string:
    mov ah,9
    lea dx,new_line
    int 21h
    ;dec length
    mov cx,0      ; current left bound
    jmp outer_loop
inner_loop:
    cmp si,length
    je inner_iteration_over
    cmp [buff+si],0Dh
    je after_change_min
    mov bh,[buff+si]
    mov bl,[buff+di]
    cmp bh,bl
    jl change_min
after_change_min:
    inc si
    jmp inner_loop
```

```

outer_loop:
    mov di,cx      ; min element index
    mov si,cx      ; i
    jmp inner_loop
inner_iteration_over:
    cmp di,length
    je set_index_registers
    mov si,cx
    mov al,[buff+si]
    mov bl,[buff+di]
    mov [buff+si],bl
    mov [buff+di],al
    inc cx        ; j
    mov di,cx
    mov si,cx
    jmp outer_loop
change_min:
    mov di,si
    jmp after_change_min
output:
    lea dx,new_line
    mov ah,9
    int 21h
    ;lea dx,buff
    ;mov ah,9
    ;int 21h
    mov si,0
output_by_symbol:
    cmp si,length
    je final
    mov ah,02h
    mov dl,[buff+si]
    int 21h
    inc si
    jmp output_by_symbol
set_index_registers:
    mov si,0
    mov di,0
    ;inc length
    jmp output_by_symbol
error:
    mov ah,9
    lea dx,new_line
    int 21h
    mov ah,9
    lea dx,error_msg
    int 21h
    jmp final
final:
    hlt

```

```

read_length:
    mov ax,@data
    mov ds,ax
    mov ah,9
    mov dx,offset msg
    int 21h
    mov ah,0Ah
    mov dx,offset string
    int 21h
    call string2number
proc string2number
    mov si,offset string + 1
    mov cl,[si]
    mov ch,0
    add si,cx
    mov bx,0
    mov bp,1
repeat:
    mov al,[si]
    sub al,48
    cmp al,0
    jl error
    cmp al,9
    ja error
    mov ah,0
    mul bp
    add bx,ax
    mov ax,bp
    mov bp,10
    mul bp
    mov bp,ax
    dec si
    loop repeat
    ret
endp

.data
buff db 200 dup('$')
new_line db 0Ah,0Dh,'$'
length dw ?
error_msg db 'Wrong input$'
msg db 'Enter length of a string: $'
string db 4 ;MAX NUMBER OF CHARACTERS ALLOWED (3).
    db ? ;NUMBER OF CHARACTERS ENTERED BY USER.
    db 5 dup (?) ;CHARACTERS ENTERED BY USER.

```

registers		
	H	L
AX	00	00
BX	00	00
CX	01	EC
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

Рисунок 1 – Значения регистров программы перед выполнением

```

BUFF      '$', '$', '$', '$', '$', '$', '$', '$', '$',
NEW_LINE      0Ah, 0Dh
LENGTH  0000h
ERROR_MSG      'w', 'r', 'o', 'n', 'g', ' ', 'i', ' '
MSG      'E', 'n', 't', 'e', 'r', ' ', 'l', 'e', 'n', 'g', 't', 'h', ' ', 'o', 'f', ' ', 'a', ' ', 's', 't'
STRING  '♦', NULL, NULL, NULL, NULL, NULL, NULL, 'P'

```

Рисунок 2 – Значение переменных программы перед выполнением

registers		
	H	L
AX	02	79
BX	79	79
CX	00	08
DX	02	79
CS	0700	
IP	01AA	
SS	0700	
SP	FFFE	
BP	000A	
SI	0008	
DI	0000	
DS	0700	
ES	0700	

Рисунок 3 – Значения регистров программы после выполнения

```

BUFF      'e', 'i', 'q', 'r', 't', 'u', 'w', 'y', '$',
NEW_LINE      0Ah, 0Dh
LENGTH  0008h
ERROR_MSG      'w', 'r', 'o', 'n', 'g', ' ', 'i',
MSG      'E', 'n', 't', 'e', 'r', ' ', 'l', 'e', 'n', 'g', 't', 'h', ' ', 'o', 'f', ' ', 'a', ' ', 's', 't',
STRING  '♦', '©', '8', 'f', NULL, NULL, NULL, 'P',

```

Рисунок 4 – значение переменных после выполнения программы после выполнения



Рисунок 5 – окно для работы с программой

**Выводы:** в результате лабораторной работы была выполнена одна задача с использованием команд JMP для безусловного перехода к метке подпрограммы, JSR для безусловного перехода к подпрограмме, RTS для возвращения из подпрограммы, BEQ для условного перехода, CMPX для сравнения.