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.

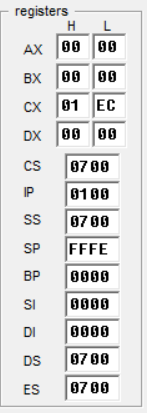


Рисунок 1 – Значения регистров программы перед выполнением

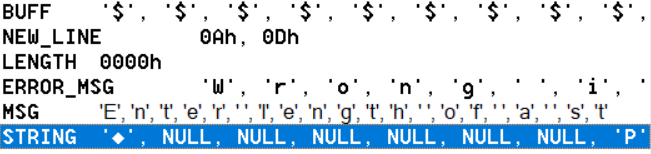


Рисунок 2 – Значение переменных программы перед выполнением

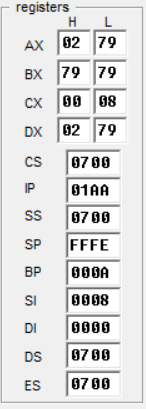


Рисунок 3 – Значения регистров программы после выполнения

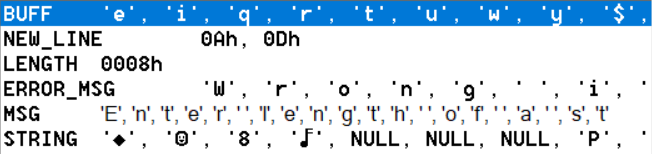


Рисунок 4 – значение переменных после выполнения программы после выполнения

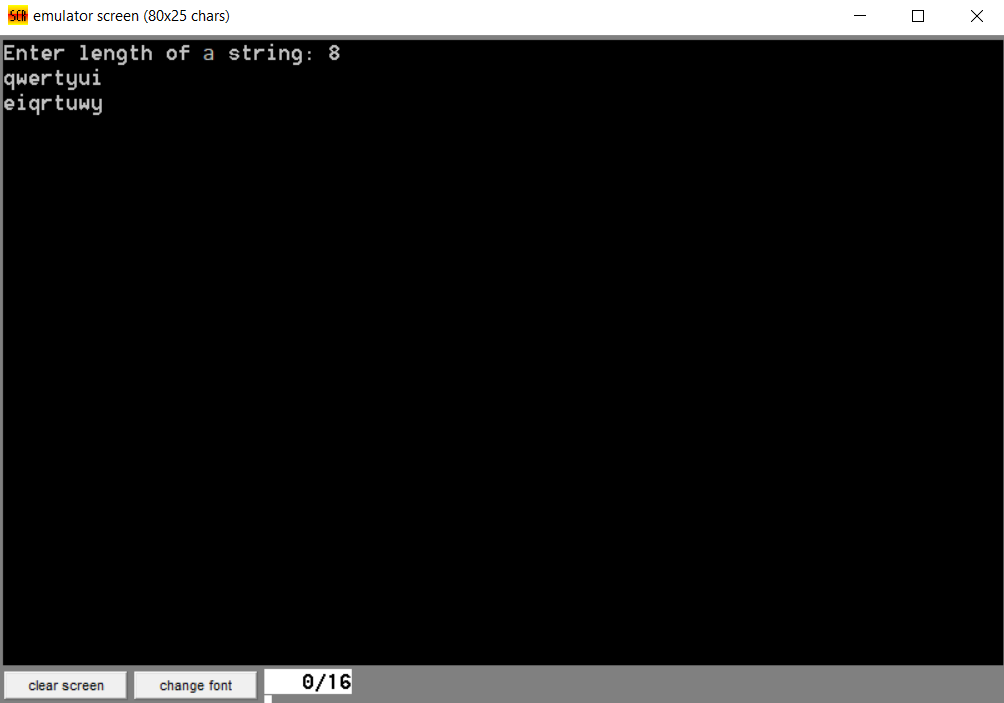


Рисунок 5 – окно для работы с программой

**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.