Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Конструирование программ

Отчёт по лабораторной работе №6

ОБРАБОТКА СИМВОЛЬНЫХ ДАННЫХ

Студент Кончик Д.С.

Принимающий Романюк М.В.

Минск 2022

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc118909172)

[2 Задание 4](#_Toc118909173)

[3 Результат выполнения 9](#_Toc118909174)

[4 Вывод 10](#_Toc118909175)

**1 ЦЕЛЬ РАБОТЫ**

1. Ознакомиться с директивами определения данных;
2. изучить команды пересылки данных и передачи управления;
3. изучить строчные операции и прерывания консольного ввода-вывода высокого уровня.

2 ЗАДАНИЕ

Вариант 12. Вставить в строке перед словом другое заданное слово.

Листинг 1 ­– Код программы

; Распечатать строку [Макрос]

printStr macro str

mov ah,09h ; Номер функции DOS

lea dx,str ; Смещение str в DX

int 21h ; 21 прерывание

endm

; Ввести строку [Макрос]

enterStr macro str

mov ah,0Ah ; Функция DOS (считать строку символов из STDIN в буфер)

lea dx,str ; Смещение строки в DX

int 21h

endm

; Задать в строке макисмальное количество символов для ввода [Макрос]

setMaxEnterSymbols macro str,number

lea di,str ; Смещение строки в DI

mov al,number ; Максимальное число символов для ввода

stosb ; AL в ES:DI

endm

; Вычислить размер введенной строки [Макрос]

; [Занести в str\_size второй байт str]

calcSizeOfEnteredStr macro str\_size,str

lea si,str ; Смещение str в SI

add si,01d ; SI++

lodsb ; DS:SI в AL

mov str\_size,al ; AL в str\_size

endm

; Отформатировать строку [Макрос]

; [Убрать первые два байта, добавить в конец '$']

formatStr macro str\_size,str

; Сдвигаем строку на 2 влево

lea di,str ; Смещение str в DI

mov si,di

add si,02d ; SI = DI + 2

mov CH,0h

mov CL,str\_size ; CX = str\_size

rep movsb ; Пересылка DS:SI в ES:DI,

; пока CX <> 0

; Добавление '$' в конец

lea di,str ; Смещение строки в DI

mov dh,0d

mov dl,str\_size ; DX = str\_size

add di,dx ; DI += str\_size

mov al,'$' ; Символ '$'

stosb ; AL в ES:DI

endm

.model small ; Модель памяти (код в 1 сегменте, данные и стек в DGROUP)

.code ; Сегмент кода

; Начало прогарммы

START:

mov ax,@data ; @data - идентификатор сегмента данных

mov ds,ax ; Адрес сегмента данных — в DS

mov es,ax ; Адрес сегмента данных — в ES

; Ввод и форматирование исходной строки

setMaxEnterSymbols string,string\_maxSize

printStr string\_info

enterStr string

calcSizeOfEnteredStr string\_size,string

cmp string\_size,0h ; if string\_size == 0

je stringEmpty ; строка пустая

formatStr string\_size,string

; Ввод и форматирование слова, перед которым вставить

setMaxEnterSymbols sfind,sfind\_maxSize

printStr toNewLine

printStr sfind\_info

enterStr sfind

calcSizeOfEnteredStr sfind\_size,sfind

cmp sfind\_size,0h ; if sfind\_size == 0

je stringEmpty ; строка пустая

formatStr sfind\_size,sfind

; Ввод и форматирование вставляемого слова

setMaxEnterSymbols spaste,spaste\_maxSize

printStr toNewLine

printStr spaste\_info

enterStr spaste

calcSizeOfEnteredStr spaste\_size,spaste

cmp spaste\_size,0h ; if spaste\_size == 0

je stringEmpty ; строка пустая

formatStr spaste\_size,spaste

; Со строками все нормально

jmp enteredStringsAreNormal

; Действие, если строка пустая

stringEmpty:

printStr toNewLine

printStr strEmpty

jmp end ; Прыжок в конец программы

; С введенными строками все нормально

enteredStringsAreNormal:

mov al,string\_size

sub al,sfind\_size

inc al

mov temp,al ; temp = string\_size - sfind\_size + 1

; до куда пойдет цикл for

mov si,0h ; SI выступает в роли счетчика i

for:

cmp si,0h ; if i == 0

je wordBeginning ; Этот символ – начало слова

cmp string[si-1],' ' ; if string[i-1]==' '

je wordBeginning ; Этот символ – начало слова

jmp notWordBeginning ; Иначе – не начало слова

; Символ – начало слова

wordBeginning:

mov bx,0h

mov bl,sfind\_size ; BX = sfind\_size

cmp string[bx][si], ' ' ; if string[i + sfind\_size] == ' '

je wordEnd ; Есть символ конца слова

cmp string[bx][si], '$' ; if string[i + sfind\_size] == '$'

je wordEnd ; Есть символ конца слова

jmp notWordEnd ; Иначе – символа конца слова нет

; Есть символ конца слова

wordEnd:

mov dx,0h

mov dl,sfind ; DX = sfind[0]

cmp string[si],dl ; if string[i] != sfind[0]

jne notSearchingWord ; Это слово – не искомое

push si ; Сохранение SI

mov ax,si

lea si,string

add si,ax ; В SI смещение на string[i]

lea di,sfind ; В DI смещение на sfind[0]

mov cx,0h

mov cl,sfind\_size ; в CX размер искомого слова

repe cmpsb ; Сравниваем string и sfind пока они равны

; и пока CX <> 0

pop si ; Восстановление SI

jnz notEqual ; Если подстрока со словом не равны

; Действие, если равны

mov place,al ; В place индекс, с которого начинается слово

mov found,01h ; found = true

jmp exitFor ; Выход из for

; Подстрока со словом не равны

notEqual:

; Это слово – не искомое

notSearchingWord:

; Нет символа конца слова

notWordEnd:

; Символ – не начало слова

notWordBeginning:

inc si ; SI++

mov ax,si

cmp al,temp ; Сравнить si с temp

loopnz for ; SI <> tmp -> следующая итерация

; Выход из цикла

exitFor:

cmp found,01h ; if found == true

je placeFound ; Место найдено

; Действие, если место не найдено

printStr toNewLine

printStr strPlaceNotFound

jmp placeNotFound

; Действие, если место найдено

placeFound:

mov al,string\_size

add al,spaste\_size

inc al

mov new\_string\_size,al ; new\_string\_size = string\_size + spaste\_size + 1

; Добавление '$' в конец (расширение строки)

lea di,string ; Смещение строки в DI

mov dh,0d

mov dl,new\_string\_size ; DX = new\_string\_size

add di,dx ; DI += new\_string\_size

mov al,'$' ; Символ '$'

stosb ; AL ('$') в ES:DI

std ; DF = 1

lea si,string

mov ax,0h

mov al,string\_size ; AX = string\_size

add si,ax

dec si ; SI – смещение на последний символ

; введенной строки

lea di,string

mov ax,0h

mov al,new\_string\_size ; AX = new\_string\_size

add di,ax

dec di ; DI – смещение на последний символ

; расширенной строки

; Освобождение места для введенного слова

mov ax,0h

mov al,new\_string\_size

dec al

sub al,place

mov cx,ax ; CX = new\_string\_size - 1 - place

rep movsb ; DS:SI в ES:DI пока CX <> 0

; Вставка введенного слова

cld ; DF = 0

lea si,spaste ; SI – смещение на первый символ

; вставляемого слова

lea di,string

mov ax,0h

mov al,place

add di,ax ; DI – смещение на место начала

; вставки в string

mov ax,0h

mov al,spaste\_size

mov cx,ax ; CX = spaste\_size

rep movsb ; DS:SI в ES:DI пока CX <> 0

; Вставка пробела после введенного слова

mov bx,0h

add bl,place

add bl,spaste\_size ; BX = place + spaste\_size

mov string[bx],' ' ; string[place + spaste\_size] = ' '

; Вывод результата

printStr toNewLine

printStr strResult

printStr string

placeNotFound: ; Место не найдено

end: ; Конец программы

mov ax,4C00h ; 4C (выход из программы) в AH

; 00 (успешное звершение) в AL

int 21h ; Функция DOS "Завершить программу"

.data ; Сегмент данных

; Исходная строка

string db 200 DUP(?)

string\_maxSize db equ 100

string\_size db 0

new\_string\_size db 0

string\_info db 'Source string: $'

; Строка, перед которой вставляем

sfind db 200 DUP(?)

sfind\_maxSize db equ 20

sfind\_size db 0

sfind\_info db 'Paste before: $'

; Строка, которую вставляем

spaste db 200 DUP(?)

spaste\_maxSize db equ 20

spaste\_size db 0

spaste\_info db 'Pasting word: $'

temp db 0 ; Временная переменная

place db 0 ; Место в строке, где начинается слово

; перед которым надо вставить

found db 0 ; bool, найдено ли слово

; Строка если не нашли слово, перед которым вставить

strPlaceNotFound db 'The word to insert before which was not found$'

strResult db 'Result: $' ; Результат

strEmpty db 'Entered string is empty$' ; Введенное слово пустое

toNewLine db 0Ah, 0Dh, '$' ; Перехода на новую строку (v,<)

.stack 100h ; Сегмент стека (256 байт)

end START ; Конец программы

3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

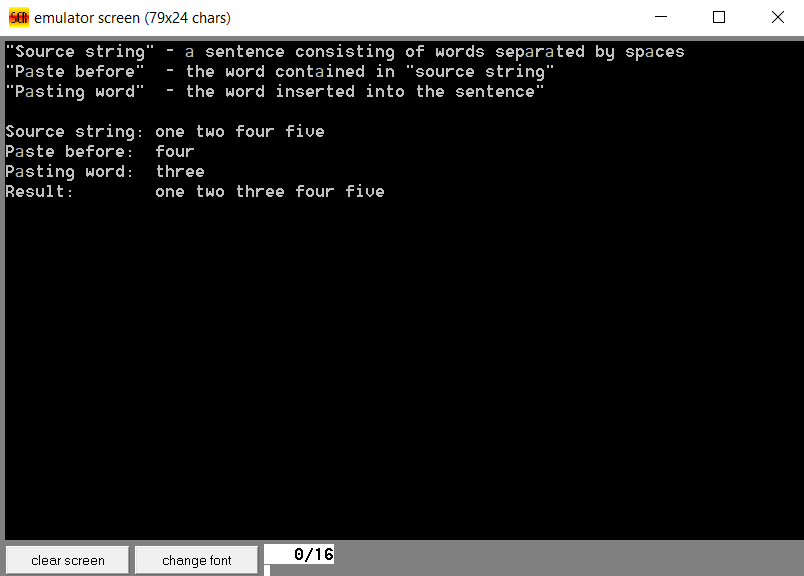


Рисунок 1 – Результат работы программы

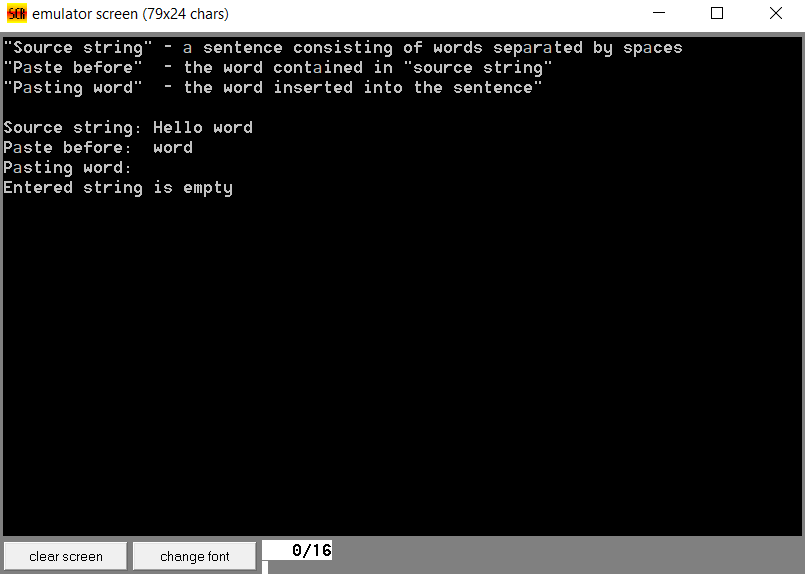


Рисунок 2 – Обработка некорректного ввода

4 ВЫВОД

Таким образом, была создана программа, принимающая на ввод исходную строку – предложение, состоящее из слов, разделенных пробелами, и два слова. Далее происходит вставка второго введенного слова перед первым в исходной строке и вывод результата.