

НИУ ВШЭ ФКН  
Микропроект №1

Иванова Виолетта Александровна  
Студентка БПИ-195(1)

**Текст задания:**

Вариант 11. Разработать программу, которая меняет на обратный порядок следования символов каждого слова в ASCII-строке символов.

## Применённые методы:

Назовём “словом” последовательность символов, которая не включает пробелы. Ввод в программу строки выполняется с помощью командной строки.

Программа действует по следующему алгоритму:

Высчитываются указатели на начало строки и конец очередного слова. Вычисляется длина слова вычитанием адресов начала строки и конца слова. Выполняется поворот слова, меняя значения соответственных символов. Запоминается результат поворота. Происходит очистка регистров. Программа повторяется, пока в строке не будут обработаны все слова.

Для получения вводимой строки использована функция `GetCommandLine`.

Длину строки определяем с помощью функции `lstrlen`.

Для обмена значений символов на соответствующих местах (процесс поворота) используется команда `xchg`.

Для чтения каждого следующего символа используется команда `lodsrb`.

Программу следует запускать через командную строку в формате: `reverse_words.exe [пользовательский ввод]`.

Программа предусматривает ввод строки любой длины, помещающейся в командную строку.

## Примеры:

```
C:\Users\Vetta\OneDrive - НИУ Высшая школа экономики\2 курс\abc\asm>reverse_words.exe 1
Source string:
1
Reversed words:
1
```

```
C:\Users\Vetta\OneDrive - НИУ Высшая школа экономики\2 курс\abc\asm>reverse_words.exe abc
Source string:
abc
Reversed words:
cba
```

```
C:\Users\Vetta\OneDrive - НИУ Высшая школа экономики\2 курс\abc\asm>reverse_words.exe abc cba acb bca
Source string:
abc cba acb bca
Reversed words:
cba abc bca acb
```

## Текст программы:

```
format PE console
entry start

include 'include\win32a.inc'

;-----SECTION IMPORT FUNCS-----
section '.idata' import data readable

library kernel,'kernel32.dll',\
        msvcrt,'msvcrt.dll'

import kernel,\
        GetCommandLine,'GetCommandLineA',\
        strlen,'strlenA',\
        ExitProcess,'ExitProcess'

import msvcrt, printf, 'printf', scanf, 'scanf', getch, '_getch'

;-----SECTION DATA DECLARATION-----
section '.data' data readable writeable

resfmt db 'Source string:', 13, 10
        db '%s', 13, 10
        db 'Reversed words:', 13, 10
        db '%s', 0

revtxt db 256 dup(0)

;-----SECTION CODE-----
section '.code' code readable executable

start:
; 1) read input args
    call getCmdLines

; 2) start word by word reversing
    call serchBeginOfNewWord

; 3) end program and print result
printResult:
    push revtxt
    push ebx
    push resfmt
    call [printf]
    call [getch]
    push 0
    call [ExitProcess]

;-----GET CMD LINE PROCERURES-----

getCmdLines:
    stdcall [GetCommandLine]    ; get cmd line args
    mov edi, eax
    ccall [strlen], eax        ; getting cmd line len
    mov ebx,eax
    mov al, ' '                ; space control
    mov ecx, ebx
    repne scasb                ; search for space or end of line
    jmp readWord

readWord:
    mov ebx,edi                ; adress of beginning
    mov esi,edi
```

```

        mov edi,revtxt
        mov edx,ecx                ; remembering numb simbols left
        jcxz skip                  ; if nothing
        rep movsb

skip:
        mov esi,revtxt

;-----WORD PROCESSING-----

serchBeginOfNewWord:
        lodsb                      ; read simbol
        cmp al, 0                  ; if end of line
        jz printResult
        cmp al, ' '                ; if not end of word
        jnz saveBeginningAdress
        jmp serchBeginOfNewWord

saveBeginningAdress:
        dec esi
        mov edx, esi              ; adress of word beginning

serchEndOfWord:
        lodsb                      ; read simbol
        cmp al, 0                  ; if end of line
        jz ifWordEnded
        cmp al, ' '                ; if not end of word
        jz ifWordEnded
        jmp serchEndOfWord

ifWordEnded:
        dec esi                    ; previous simbol
        mov eax,esi                ; word end adress
        sub eax,edx                ; get len of word
        ccall reverseWord, edx, eax
        jmp serchBeginOfNewWord

;-----REVERSING ONE WORD-----

reverseWord:
        push ebp
        mov ebp, esp
        push esi
        push edi
        mov esi, [ebp+8]           ; adress of reversed
        mov ecx, [ebp+12]          ; sourse len
        lea edx, [esi+ecx-1]       ; adress end sourse
        shr ecx, 1
        jecxz cleanRegisters      ; check len

reverseSimbols:
        lodsb                      ; read new simbol
        xchg al, [edx]             ; reverse
        mov [esi-1], al
        dec edx
        loop reverseSimbols       ; make reverse

cleanRegisters:
        pop edi
        pop esi
        pop ebp
        ret

```