МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №6**

**з дисципліни «Архітектура комп'ютера»**

## на тему:« Організація введення/виведення інформації з консолі в мові Ассемблера. Обробка масивів символів в мові Ассемблера..»

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2020

**Тема.** Організація введення/виведення інформації з консолі в мові Ассемблера. Обробка масивів символів в мові Ассемблера.

**Мета.** Вивчити засоби мови Ассемблера для введення/виведення символьної інформації. Набути практичних навичок з обробки масивів мовою Ассемблера.

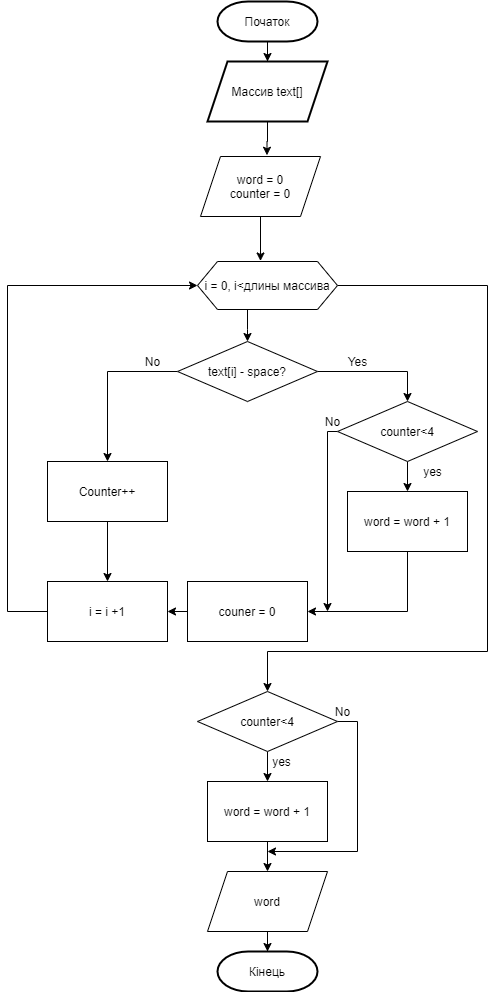
Варіант 4

**1. Завдання:** розробити програму, яка вводить з консолі рядок символів, обробляє його відповідно до обраного варіанту завдання і виводить на консоль результат обробки.

Порахувати кількість слів, довжина яких не перевищує 3 символи

**2. Вимоги до програми:**  
- вхідні дані вводяться з клавіатури;  
- результати роботи програми виводяться на екран.

**3. Алгоритм**

****

**4. Тести**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва тесту | Вхідні дані | Результат |
| 1 | Слова з довільною кількістю символів | A cat with nine lives | 2 |
| 2 | Усі слова >3 символів | Keyboard computer shark mirror | 0 |
| 3 | Усі слова <=3 символів | I you he she it we | 6 |

**5. Лістинг**

Turbo Assembler Version 3.0 12/06/20 13:40:19 Page 1

ak6.asm

1 ;=======================================================================================

2 ;Розробити програму, яка вводить з консолі рядок +

3 символів,

4 ;обробляє його відповідно до обраного варіанту +

5 завдання і виводить на консоль результат обробки.

6 ;Порахувати кількість слів, довжина яких не +

7 перевищує 3 символи.

8 ;Вхідні дані: рядок символів

9 ;Вихідні дані: кількість слів, символів у яких менше +

10 4-х

11 ;=======================================================================================

12 ;=================сегмент стека=================

13 0000 stk segment stack

14 0000 80\*(??) db 128 dup(?)

15 0080 stk ends

16

17 ;======================сегмент даних=================

18 0000 data segment para public 'data'

19 0000 50 buffer db 80

20 0001 ?? text db ?

21 0002 50\*(20) string db 80 dup (' ')

22 0052 45 6E 74 65 72 20 73+ msg\_prompt db 'Enter string : $'

23 74 72 69 6E 67 20 3A+

24 20 24

25 0062 52 65 73 75 6C 74 20+ msg\_result db 'Result : $'

26 3A 20 24

27 006C 00 slovo db 0

28 006D 0A 0D 24 crlf db 0ah, 0dh, '$'

29 0070 data ends

30

31 ;==========================сегмент команд========================

32 0000 code segment para public 'code'

33 assume cs:code, ds:data, ss:stk

34 0000 start:

35 0000 B8 0000s mov ax, data

36 0003 8E D8 mov ds, ax

37 ;=========================основна программа======================

38 0005 in\_text:

39 0005 BA 0052r lea dx, msg\_prompt ;виведення запрошення

40 0008 B4 09 mov ah, 09h

41 000A CD 21 int 21h

42 000C BA 0000r lea dx, buffer ;запис рядка в буфер

43 000F B4 0A mov ah, 0ah

44 0011 CD 21 int 21h

45 0013 F6 06 0001r FF test text, 0FFh ;якщо введений рядок порожній,

46 0018 74 EB jz in\_text ;то перехід на повторне +

47 введення рядка

48 001A BA 006Dr lea dx, crlf

49 001D B4 09 mov ah, 09h

50 001F CD 21 int 21h

51 0021 33 C9 xor cx, cx ;cx = 0

52 0023 8A 0E 0001r mov cl, text ;лічильник циклу

53 0027 33 F6 xor si, si ;індекс

54 0029 33 DB xor bx, bx ;bx = counter

55 002B while\_begin:

56 002B 80 BC 0002r 20 cmp string[si], 20h ;порівння поточного символу з +

57 пробілом

\_Turbo Assembler Version 3.0 12/06/20 13:40:19 Page 2

ak6.asm

58 0030 75 0E jne not\_space ;якщо символ не пробіл, +

59 перейти до мітки not\_space

60 0032 83 FB 03 cmp bx, 03h ;інакше порівняти counter та 3

61 0035 7F 04 jg next ;якщо counter > 3, перейти до +

62 мітки next

63 0037 FE 06 006Cr inc slovo ;збільшити лічильник слів

64 003B next:

65 003B 33 DB xor bx, bx ;counter = 0

66 003D 46 inc si ;збільшити індекс

67 003E E2 EB loop while\_begin ;якщо не кінець рядка, перейти +

68 на початок циклу

69 0040 not\_space:

70 0040 43 inc bx ;збыльшити counter на 1

71 0041 46 inc si ;збільшити індекс на 1

72 0042 E2 E7 loop while\_begin

73 0044 83 FB 03 cmp bx, 03h ;порівняти counter та 3

74 0047 7F 04 jg next2 ;якщо counter > 3, перейти до +

75 мітки next2

76 0049 FE 06 006Cr inc slovo

77 004D next2:

78 004D BA 0062r lea dx, msg\_result ;вивід результату на екран

79 0050 B4 09 mov ah, 09h

80 0052 CD 21 int 21h

81 0054 A0 006Cr mov al, slovo

82 0057 05 0030 add ax, 30h

83 005A 8A D0 mov dl, al

84 005C B4 02 mov ah, 02h

85 005E CD 21 int 21h

86 ;=====================кінець програми========================

87 0060 endprog:

88 0060 B8 4C00 mov ax, 4C00h

89 0063 CD 21 int 21h

90 0065 code ends

91 end start

\_Turbo Assembler Version 3.0 12/06/20 13:40:19 Page 3

Symbol Table

Symbol Name Type Value

??DATE Text "12/06/20"

??FILENAME Text "ak6 "

??TIME Text "13:40:19"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text AK6

@WORDSIZE Text 2

BUFFER Byte DATA:0000

CRLF Byte DATA:006D

ENDPROG Near CODE:0060

IN\_TEXT Near CODE:0005

MSG\_PROMPT Byte DATA:0052

MSG\_RESULT Byte DATA:0062

NEXT Near CODE:003B

NEXT2 Near CODE:004D

NOT\_SPACE Near CODE:0040

SLOVO Byte DATA:006C

START Near CODE:0000

STRING Byte DATA:0002

TEXT Byte DATA:0001

WHILE\_BEGIN Near CODE:002B

Groups & Segments Bit Size Align Combine Class

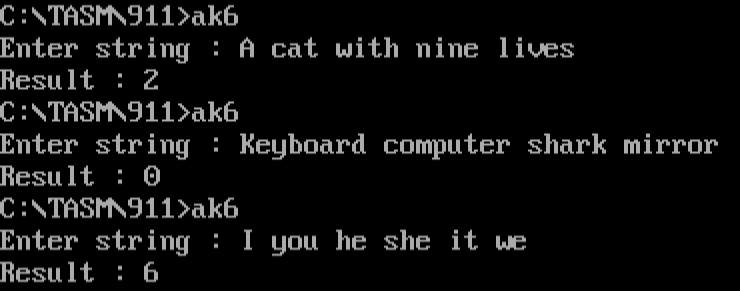
CODE 16 0065 Para Public CODE

DATA 16 0070 Para Public DATA

STK 16 0080 Para Stack

\_

**6. Результати тестувань**



На скріншоті представлені результати тестувань. У всіх трьох випадках программа спрацювала вірно.

**7. Висновок**

Під час виконання лабораторної роботи, працював з організацією вводу та виводу мовою ассемблера. Для вводу та виводу символа з клавіатури використовують функцію двадцять першого переривання 01h та 02h відповідно. Для введення символьного рядка в массив необхідно підготувати буфер вводу у сегменті данних. Потім викоритстовувати функцію 0ah та 09h для вводу/виводу рядка. Для обробки інформації в масиві використовував цикл for та регістр si для доступу до елементів массиву.