МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський 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 та виведення інформації в консолі на рівні Ассемблера. Обробка масивів чисел в мові Ассемблера.»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2019

**Лабораторна робота №6.1**

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

**Мета**. Набути практичних навичок розробки програм циклічної структури.

**Постановка задачи**

1. Загальне завдання

Розробити програму, яка вводить з консолі рядок символів, обробляє його відповідно до обраного варіанту завдання і виводить на консоль результат обробки.

2. Індивідуальне завдання

|  |  |
| --- | --- |
| 7 | Порахувати кількість слів, коди символів яких утворюють монотонно зростаючу послідовність. |

3. Вимоги до програми:

- вхідні дані вводяться з клавіатури;

- результати роботи програми виводяться на екран.

4. Вимоги до вхідних даних:

- рядок символів довільної довжини, але не більше 80 символів, який може містити пробіли, великі та малі літери англійського алфавіту, цифри (перевіряти зміст рядка не треба, рядок завжди задається правильно);

- рядок починається з символа;

- словом вважається будь-яка послідовність символів, що відрізняється від пробілу; - слова відділяються одним пробілом;

- окрім рядка введенню з клавіатури підлягає додаткова інформація (символи).

5. Вимоги до тексту програми:

- коментарі щодо призначення програми, її вхідних і вихідних даних;

- коментарі щодо призначення кожного блоку програми, дій окремих команд для пояснення алгоритму;

- самодокументованість коду: всі ідентифікатори повинні мати назви, що відповідають суті змінних і міток.

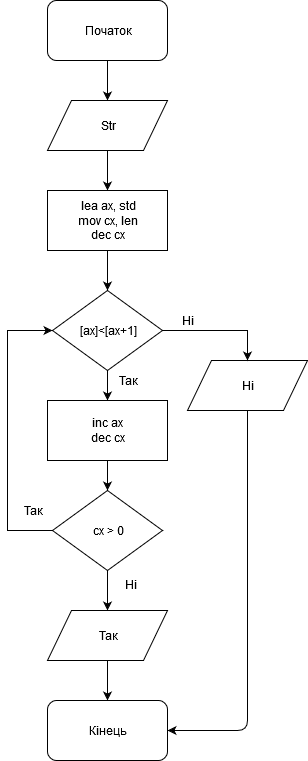
**Розробка тестів**

1. Тести

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вхідні дані | Очікувані результати | Підтвердження результату |
| 1 | 167 4 6 432 09 | 4 | + |
| 2 | (порожньо) | 0 | + |
| 3 | 125 | 1 | + |
| 4 | 975 75 52 | 0 | + |

**Алгоритм програм**

1.Блок-схема



2.Текст програми

sseg segment stack 'stack'

db 128 dup(?)

sseg ends

dseg segment 'data'

base dw 10

minus dw -1

txtin db 'Enter text: $'

txtout db 'Amount of words with mono-growing characters: $'

bufsize db 255

bufread db 0

buf db 255 dup('$')

dseg ends

cseg segment 'code'

assume cs:cseg, ds:dseg, ss:sseg

textin proc near

mov ah, 0Ah

int 21h

ret

textin endp

textout proc near

mov ah, 09h

int 21h

ret

textout endp

chout proc near

mov ah, 02h

int 21h

ret

chout endp

iteratechs proc near

push ax

mov bx, 0

cmp cx, 0

je return

chloop:

mov ax, word ptr [di]

cmp ah, ' '

je space\_found

cmp ah, 0Dh

je endline\_found

cmp ah, al

ja endif\_after

mov bx, 1

endif\_after:

inc di

loopne chloop

jmp return

space\_found:

inc di

dec cx

jmp return

endline\_found:

dec cx

return:

pop ax

ret

iteratechs endp

iteratewords proc near

push bx

wordloop:

call iteratechs

inc di

cmp bx, 1

je endif\_seqgrow

inc ax

endif\_seqgrow:

loop wordloop

pop bx

ret

iteratewords endp

numout proc near

push bx

push dx

push cx

cmp ax, 0

jge output\_number\_unsigned

imul minus

xchg ax, bx

mov dl, '-'

call chout

xchg ax, bx

output\_number\_unsigned:

mov bx, 0

mov cx, 0

output\_number\_loop:

cmp ax, 0

cwd

idiv base

add dl, '0'

push dx

inc bx

cmp ax, 0

loopne output\_number\_loop

mov cx, bx

output\_stack:

pop dx

call chout

loop output\_stack

pop cx

pop dx

pop bx

ret

numout endp

start:

mov ax, dseg

mov ds, ax

lea dx, txtin

call textout

lea dx, bufsize

call textin

mov bx, 0

mov ch, 0

mov cl, bufread

inc cx

mov ax, 0

cmp cl, 1

je outputans

lea di, buf

call iteratewords

outputans:

push ax

mov dl, 0Ah

call chout

mov dl, 0Dh

call chout

lea dx, txtout

call textout

pop ax

call numout

mov dl, 0Ah

call chout

mov dl, 0Dh

call chout

mov ah, 4Ch

int 21h

cseg ends

end start

3.Лістинг

Turbo Assembler Version 3.0 01/22/20 16:58:22 Page 1

lab6.asm

1 0000 sseg segment stack 'stack'

2 0000 80\*(??) db 128 dup(?)

3 0080 sseg ends

4

5 0000 dseg segment 'data'

6

7 0000 000A base dw 10

8 0002 FFFF minus dw -1

9 0004 45 6E 74 65 72 20 74+ txtin db 'Enter text: $'

10 65 78 74 3A 20 24

11 0011 41 6D 6F 75 6E 74 20+ txtout db 'Amount of words with mono-growing characters: $'

12 6F 66 20 77 6F 72 64+

13 73 20 77 69 74 68 20+

14 6D 6F 6E 6F 2D 67 72+

15 6F 77 69 6E 67 20 63+

16 68 61 72 61 63 74 65+

17 72 73 3A 20 24

18 0040 FF bufsize db 255

19 0041 00 bufread db 0

20 0042 FF\*(24) buf db 255 dup('$')

21

22 0141 dseg ends

23

24 0000 cseg segment 'code'

25 assume cs:cseg, ds:dseg, ss:sseg

26

27 0000 textin proc near

28 0000 B4 0A mov ah, 0Ah

29 0002 CD 21 int 21h

30 0004 C3 ret

31 0005 textin endp

32

33 0005 textout proc near

34 0005 B4 09 mov ah, 09h

35 0007 CD 21 int 21h

36 0009 C3 ret

37 000A textout endp

38

39 000A chout proc near

40 000A B4 02 mov ah, 02h

41 000C CD 21 int 21h

42 000E C3 ret

43 000F chout endp

44

45 000F iteratechs proc near

46 000F 50 push ax

47 0010 BB 0000 mov bx, 0

48 0013 83 F9 00 cmp cx, 0

49 0016 74 1F je return

50

51 0018 chloop:

52 0018 8B 05 mov ax, word ptr [di]

53 001A 80 FC 20 cmp ah, ' '

54 001D 74 12 je space\_found

55 001F 80 FC 0D cmp ah, 0Dh

56 0022 74 12 je endline\_found

57 0024 3A E0 cmp ah, al

58 0026 77 03 ja endif\_after

59 0028 BB 0001 mov bx, 1

60 002B endif\_after:

61 002B 47 inc di

62 002C E0 EA loopne chloop

63 002E EB 07 90 jmp return

64 0031 space\_found:

65 0031 47 inc di

66 0032 49 dec cx

67 0033 EB 02 90 jmp return

68 0036 endline\_found:

69 0036 49 dec cx

70

71 0037 return:

72 0037 58 pop ax

73 0038 C3 ret

74 0039 iteratechs endp

75

76 0039 iteratewords proc near

77 0039 53 push bx

78

79 003A wordloop:

80 003A E8 FFD2 call iteratechs

81 003D 47 inc di

82 003E 83 FB 01 cmp bx, 1

83 0041 74 01 je endif\_seqgrow

84 0043 40 inc ax

85

86 0044 endif\_seqgrow:

87 0044 E2 F4 loop wordloop

88 0046 5B pop bx

89 0047 C3 ret

90 0048 iteratewords endp

91

92 0048 numout proc near

93 0048 53 push bx

94 0049 52 push dx

95 004A 51 push cx

96

97 004B 3D 0000 cmp ax, 0

98 004E 7D 0B jge output\_number\_unsigned

99 0050 F7 2E 0002r imul minus

100 0054 93 xchg ax, bx

101 0055 B2 2D mov dl, '-'

102 0057 E8 FFB0 call chout

103 005A 93 xchg ax, bx

104 005B output\_number\_unsigned:

105 005B BB 0000 mov bx, 0

106 005E B9 0000 mov cx, 0

107 0061 output\_number\_loop:

108 0061 3D 0000 cmp ax, 0

109 0064 99 cwd

110 0065 F7 3E 0000r idiv base

111 0069 80 C2 30 add dl, '0'

112 006C 52 push dx

113 006D 43 inc bx

114 006E 3D 0000 cmp ax, 0

115 0071 E0 EE loopne output\_number\_loop

116 0073 8B CB mov cx, bx

117 0075 output\_stack:

118 0075 5A pop dx

119 0076 E8 FF91 call chout

120 0079 E2 FA loop output\_stack

121

122 007B 59 pop cx

123 007C 5A pop dx

124 007D 5B pop bx

125 007E C3 ret

126 007F numout endp

127

128 007F start:

129 007F B8 0000s mov ax, dseg

130 0082 8E D8 mov ds, ax

131

132 0084 BA 0004r lea dx, txtin

133 0087 E8 FF7B call textout

134 008A BA 0040r lea dx, bufsize

135 008D E8 FF70 call textin

136 0090 BB 0000 mov bx, 0

137 0093 B5 00 mov ch, 0

138 0095 8A 0E 0041r mov cl, bufread

139 0099 41 inc cx

140 009A B8 0000 mov ax, 0

141 009D 80 F9 01 cmp cl, 1

142 00A0 74 06 je outputans

143 00A2 BF 0042r lea di, buf

144 00A5 E8 FF91 call iteratewords

145

146 00A8 outputans:

147 00A8 50 push ax

148 00A9 B2 0A mov dl, 0Ah

149 00AB E8 FF5C call chout

150 00AE B2 0D mov dl, 0Dh

151 00B0 E8 FF57 call chout

152 00B3 BA 0011r lea dx, txtout

153 00B6 E8 FF4C call textout

154 00B9 58 pop ax

155 00BA E8 FF8B call numout

156

157 00BD B2 0A mov dl, 0Ah

158 00BF E8 FF48 call chout

159 00C2 B2 0D mov dl, 0Dh

160 00C4 E8 FF43 call chout

161 00C7 B4 4C mov ah, 4Ch

162 00C9 CD 21 int 21h

163

164 00CB cseg ends

165 end start

Symbol Name Type Value

??DATE Text "01/22/20"

??FILENAME Text "lab6 "

??TIME Text "16:58:22"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CSEG

@FILENAME Text LAB6

@WORDSIZE Text 2

BASE Word DSEG:0000

BUF Byte DSEG:0042

BUFREAD Byte DSEG:0041

BUFSIZE Byte DSEG:0040

CHLOOP Near CSEG:0018

CHOUT Near CSEG:000A

ENDIF\_AFTER Near CSEG:002B

ENDIF\_SEQGROW Near CSEG:0044

ENDLINE\_FOUND Near CSEG:0036

ITERATECHS Near CSEG:000F

ITERATEWORDS Near CSEG:0039

MINUS Word DSEG:0002

NUMOUT Near CSEG:0048

OUTPUTANS Near CSEG:00A8

OUTPUT\_NUMBER\_LOOP Near CSEG:0061

OUTPUT\_NUMBER\_UNSIGNED Near CSEG:005B

OUTPUT\_STACK Near CSEG:0075

RETURN Near CSEG:0037

SPACE\_FOUND Near CSEG:0031

START Near CSEG:007F

TEXTIN Near CSEG:0000

TEXTOUT Near CSEG:0005

TXTIN Byte DSEG:0004

TXTOUT Byte DSEG:0011

WORDLOOP Near CSEG:003A

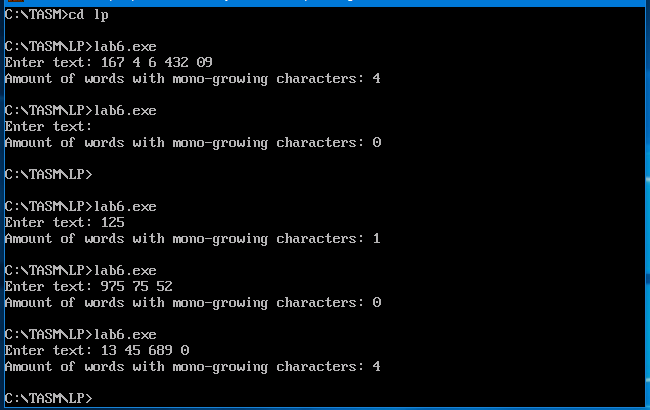
Groups & Segments Bit Size Align Combine Class

CSEG 16 00CB Para none CODE

DSEG 16 0141 Para none DATA

SSEG 16 0080 Para Stack STACK

4. Результати виконання програми



Висновок: у даній лабораторній роботі я ознайомився зорганізація введення/виведення інформації з консолі в мові Ассемблера, а також з обробкою масивів символів в мові Ассемблера. Була створена блок-схема та розроблена програма до індивідуального завдання.