**Московский государственный технический университет им. Н.Э.Баумана**

|  |  |  |
| --- | --- | --- |
| Защищено:  Большаков С.А.    " " | 2025 г. | Демонстрация ЛР: Большаков С.А.  " " 2025 г. |

**Отчет по лабораторной работе № 4 по курсу**

**Системное программирование**

**"** **Циклы и перевод символов "**

**(есть ли дополнительные требования - ДА)**

15

(количество листов)

Вариант № <**2**>

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-42Б**

(подпись)

**Афонин И.И.**

"23" марта 2025 г.

Москва, МГТУ - 2025

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[1. Цель выполнения лабораторной работы № 4 3](#_Toc193649643)

[2. Порядок и условия проведения работы № 4 3](#_Toc193649644)

[3. Описание ошибок, возникших при отладке № 4 3](#_Toc193649645)

[4. Блок-схема программы 3](#_Toc193649646)

[5. Скриншот программы KH № 4 в TD.exe 5](#_Toc193649647)

[6. Текст программы на языке Ассемблера 5](#_Toc193649648)

[7. Результаты работы программы 9](#_Toc193649649)

[8. Выводы по ЛР № 4 15](#_Toc193649650)

# Цель выполнения лабораторной работы № 4

Лабораторная работа №4 выполняется для получения навыков разработки и отладки программ на ЯП, получения базовых знаний об использовании циклов и перевода символов из одной системы в другую на Ассемблере, изучения и использования компонентов системы программирования Ассемблер (компилятора, редактора связей, отладчика) и получения навыков оформления документации по программным разработкам, реализуемым на языке.

# Порядок и условия проведения работы № 4

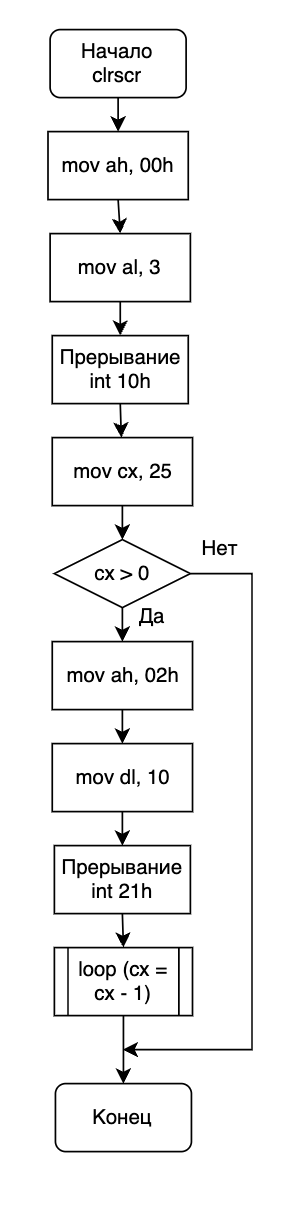
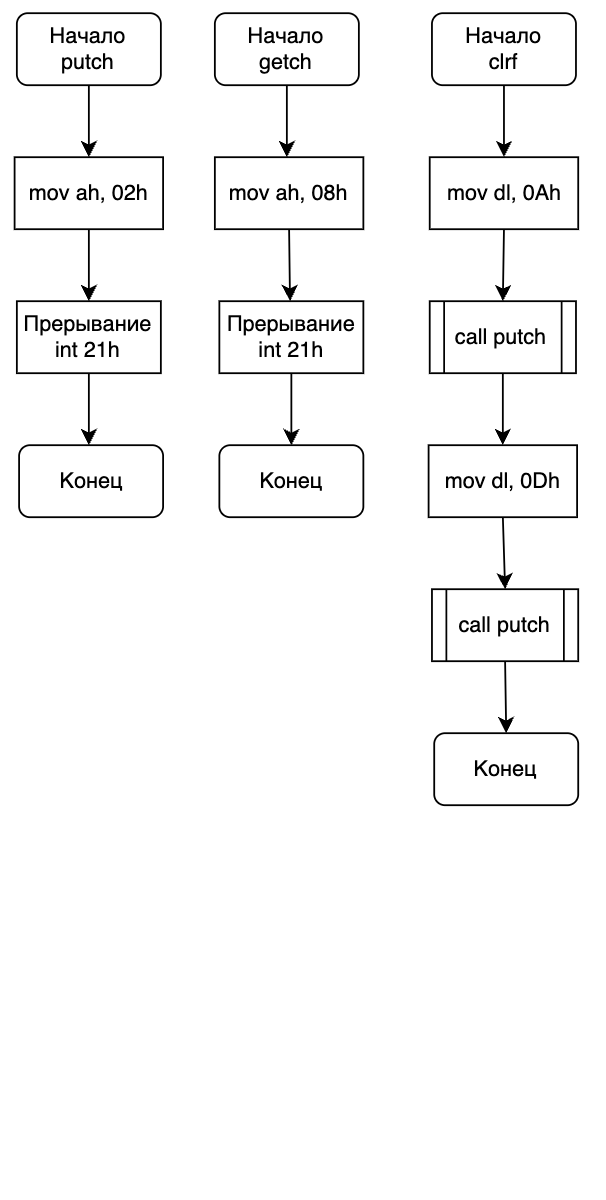
Разработать и отладить циклическую программу на языке Ассемблер для вывода на экран 20 последовательных прописных букв русского алфавита (начиная с символа “А” или другого символа, введенного с клавиатуры).

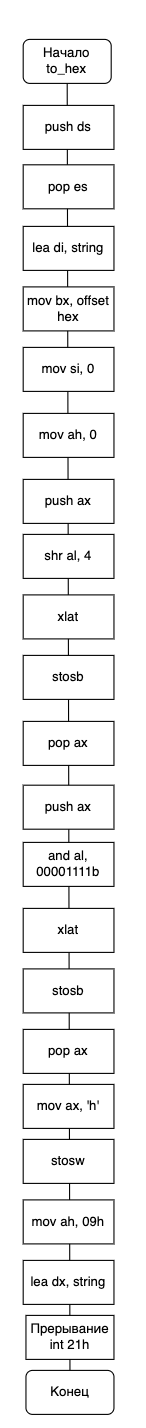
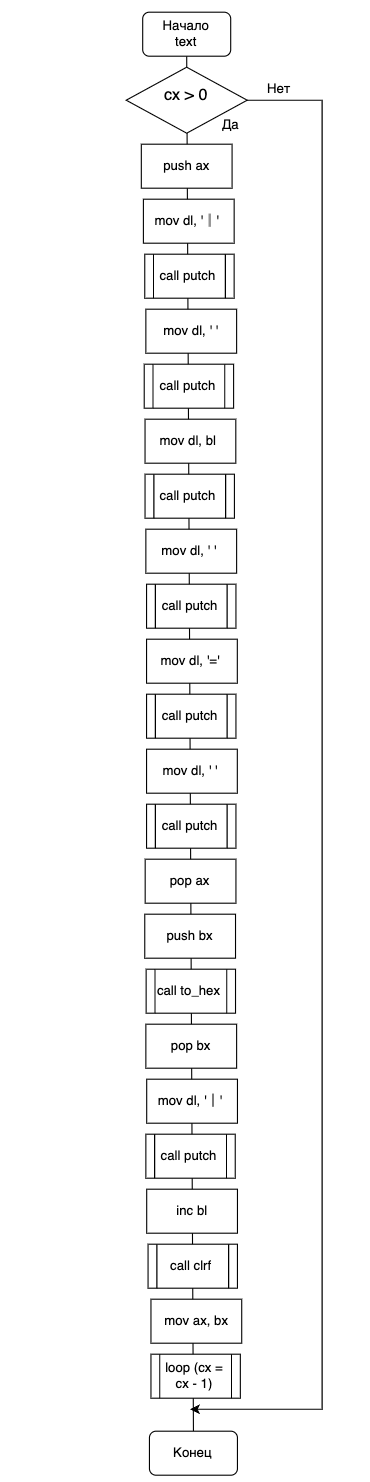
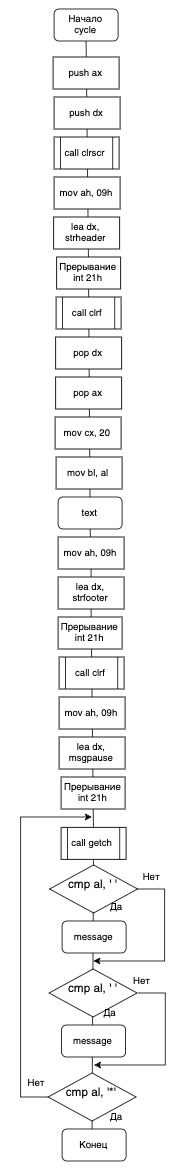
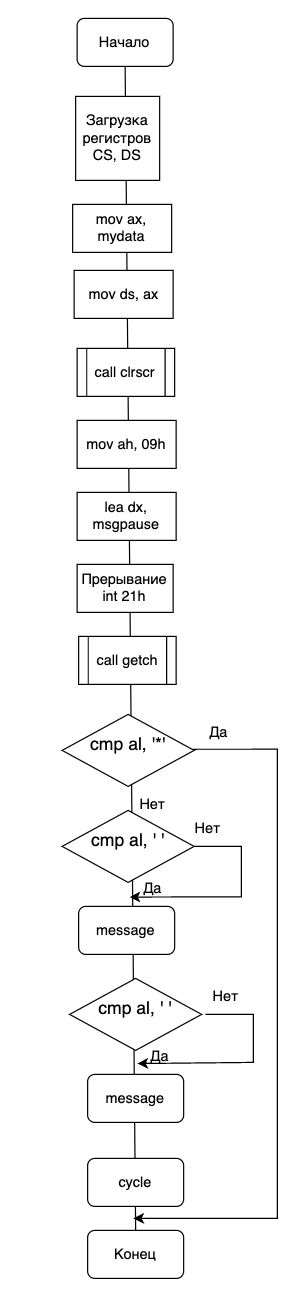
После завершения вывода букв и их кодов на экран организовать ожидание ввода любого символа с клавиатуры (нажатие клавиши).

# Описание ошибок, возникших при отладке № 4

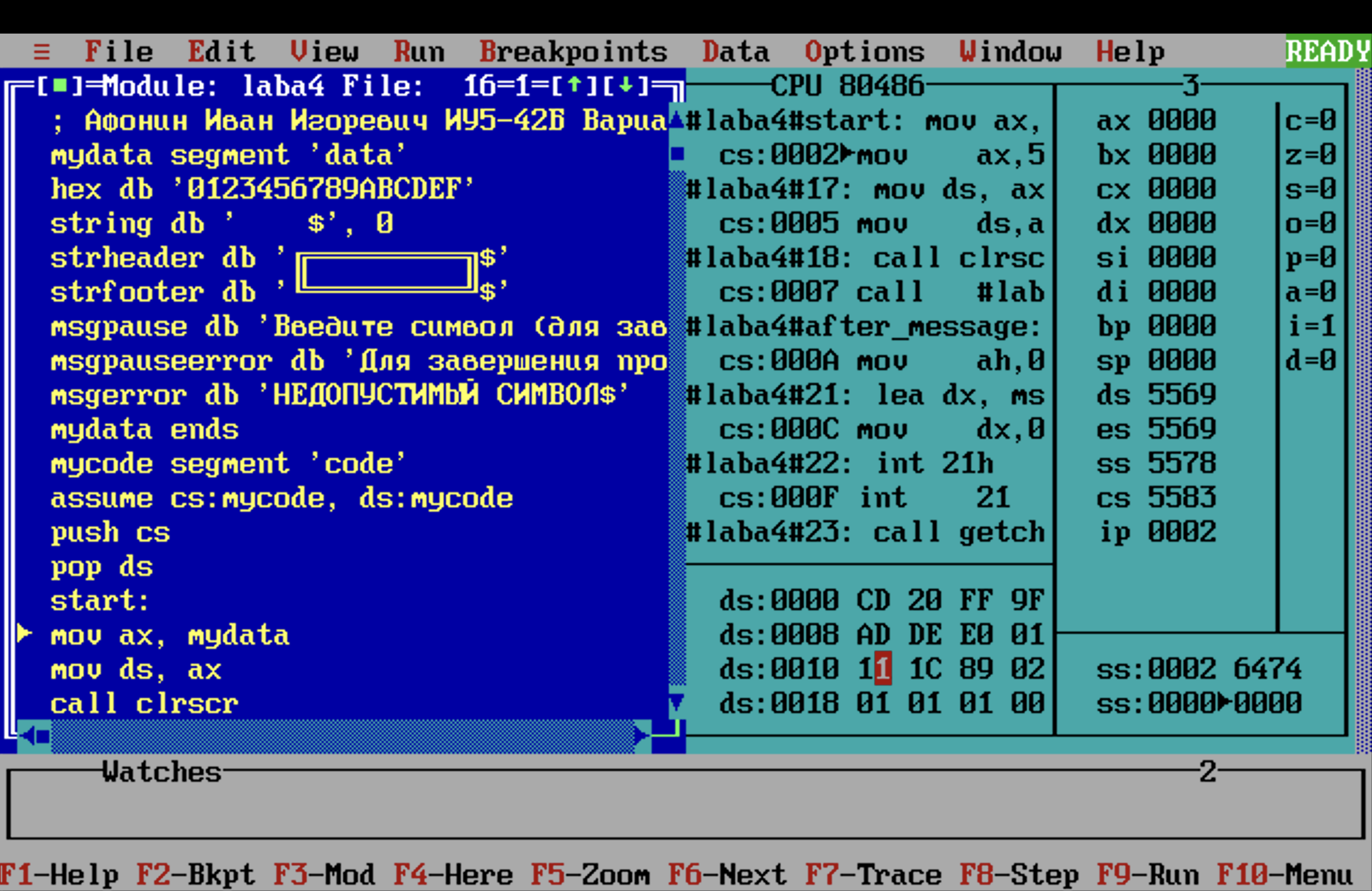
Ошибки отсутствуют.

# Блок-схема программы





# Скриншот программы KH № 4 в TD.exe



# Текст программы на языке Ассемблера

; Афонин Иван Игоревич ИУ5-42Б Вариант 2 ЛР4

mydata segment 'data'

hex db '0123456789ABCDEF'

string db ' $', 0

strheader db '╔═════════╗$'

strfooter db '╚═════════╝$'

msgpause db 'Введите символ (для завершения введите \*): $'

msgpauseerror db 'Для завершения программы нажмите любую клавишу: $'

msgerror db 'НЕДОПУСТИМЫЙ СИМВОЛ$'

mydata ends

mycode segment 'code'

assume cs:mycode, ds:mycode

push cs

pop ds

start:

mov ax, mydata

mov ds, ax

call clrscr

after\_message:

mov ah, 09h

lea dx, msgpause

int 21h

call getch

cmp al, '\*'

je program\_end

cmp al, ' '

jb message

cmp al, ' '

ja message

cycle:

push ax

push dx

call clrscr

mov ah, 09h

lea dx, strheader

int 21h

call clrf

pop dx

pop ax

mov cx, 20

mov bl, al

text:

push ax

mov dl, '║'

call putch

mov dl, ' '

call putch

mov dl, bl

call putch

mov dl, ' '

call putch

mov dl, '='

call putch

mov dl, ' '

call putch

pop ax

push bx

call to\_hex

pop bx

mov dl, '║'

call putch

inc bl

call clrf

mov ax, bx

loop text

mov ah, 09h

lea dx, strfooter

int 21h

call clrf

mov ah, 09h

lea dx, msgpause

int 21h

call getch

cmp al, ' '

jb message

cmp al, ' '

ja message

; Проверяем, какой символ ввели с клавиатуры

cmp al, '\*'

; Если ввели символ, отличный от \*, повторяем вывод символов

jne cycle

; Очистка экрана и выход

program\_end:

call clrscr

mov al, 05h

mov ah, 4ch

int 21h

message:

mov ah, 09h

lea dx, msgerror

int 21h

call clrf

jmp after\_message

; Процедура вывода символа на экран

putch proc

mov ah, 02h

int 21h

ret

putch endp

; Процедура ожидания ввода символа с клавиатуры

getch proc

mov ah, 08h

int 21h

ret

getch endp

; Процедура перевода строки

clrf proc

mov dl, 0Ah

call putch

mov dl, 0Dh

call putch

ret

clrf endp

; Процедура очистки экрана

clrscr proc

mov ah, 00h

mov al, 3

int 10h

mov cx, 25

labclr:

mov ah, 02h

mov dl, 10

int 21h

loop labclr

ret

clrscr endp

; Процедура перевода числа в шестнадцатиричное представление

to\_hex proc

push ds

pop es

lea di, string

mov bx, offset hex

mov si, 0

mov ah, 0

push ax

shr al, 4

xlat

stosb

pop ax

push ax

and al, 00001111b

xlat

stosb

pop ax

mov ax, 'h'

stosw

mov ah, 09h

lea dx, string

int 21h

ret

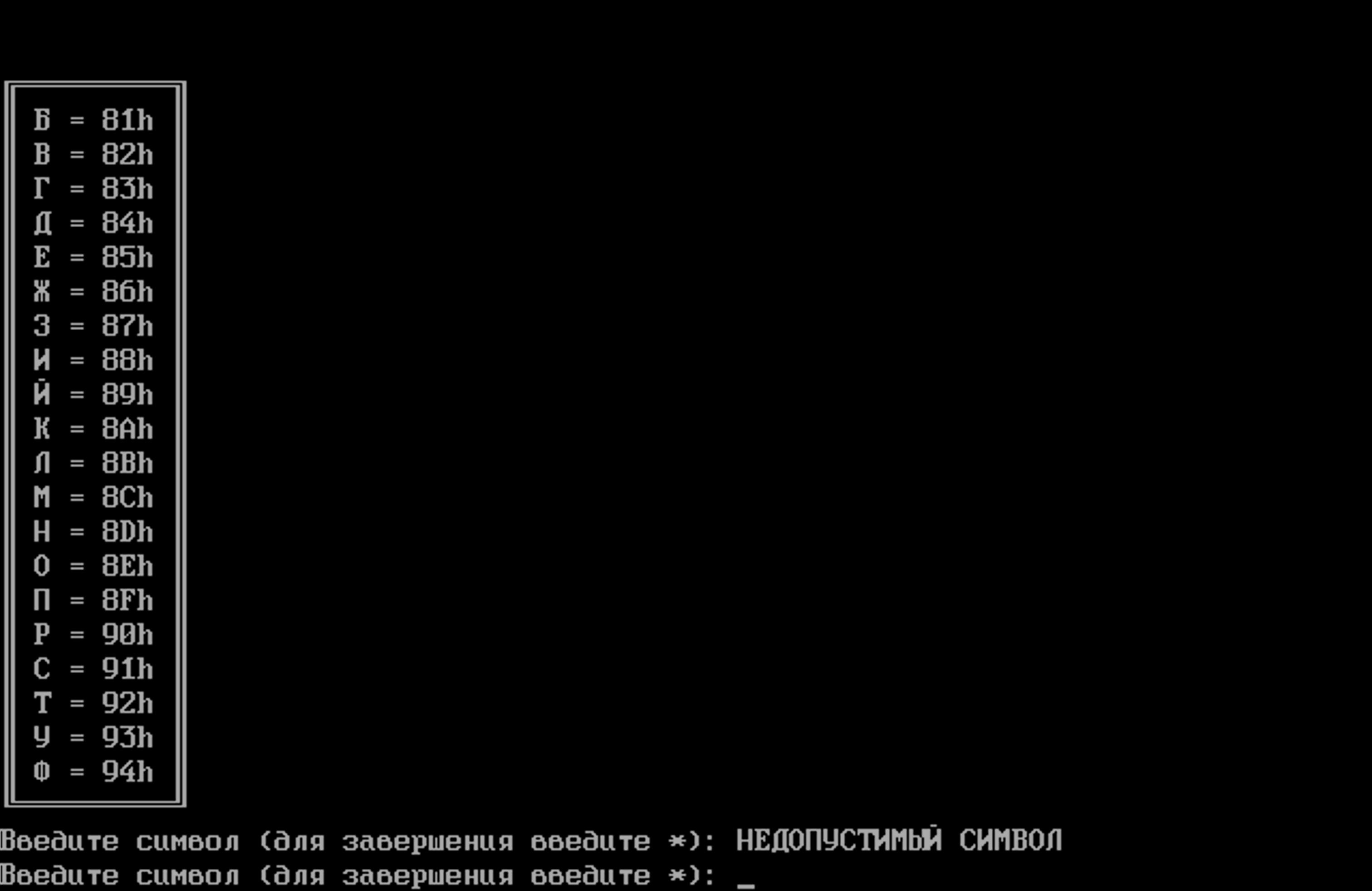
to\_hex endp

mycode ends

end start

# Результаты работы программы





LABA4.LST

Turbo Assembler Version 3.1 03/23/25 16:12:29 Page 1

laba4.asm

1 ; Афонин Иван Игоревич ИУ5-42Б Вариант 2 ЛР4

2 0000 mydata segment 'data'

3 0000 30 31 32 33 34 35 36+ hex db '0123456789ABCDEF'

4 37 38 39 41 42 43 44+

5 45 46

6 0010 20 20 20 20 24 00 string db ' $', 0

7 0016 C9 CD CD CD CD CD CD+ strheader db '╔═════════╗$'

8 CD CD CD BB 24

9 0022 C8 CD CD CD CD CD CD+ strfooter db '╚═════════╝$'

10 CD CD CD BC 24

11 002E 82 A2 A5 A4 A8 E2 A5+ msgpause db '‚Введите символ (для завершения введите \*): $'

12 20 E1 A8 AC A2 AE AB+

13 20 28 A4 AB EF 20 A7+

14 A0 A2 A5 E0 E8 A5 AD+

15 A8 EF 20 A2 A2 A5 A4+

16 A8 E2 A5 20 2A 29 3A+

17 20 24

18 005A 84 AB EF 20 A7 A0 A2+ msgpauseerror db 'Для завершения программы нажмите любую клавишу: $'

19 A5 E0 E8 A5 AD A8 EF+

20 20 AF E0 AE A3 E0 A0+

21 AC AC EB 20 AD A0 A6+

22 AC A8 E2 A5 20 AB EE+

23 A1 E3 EE 20 AA AB A0+

24 A2 A8 E8 E3 3A 20 24

25 008B 8D 85 84 8E 8F 93 91+ msgerror db 'НЕДОПУСТИМЫЙ СИМВОЛ$'

26 92 88 8C 9B 89 20 91+

27 88 8C 82 8E 8B 24

28 009F mydata ends

29 0000 mycode segment 'code'

30 assume cs:mycode, ds:mycode

31 0000 0E push cs

32 0001 1F pop ds

33 0002 start:

34 0002 B8 0000s mov ax, mydata

35 0005 8E D8 mov ds, ax

36 0007 E8 00AA call clrscr

37 000A after\_message:

38 000A B4 09 mov ah, 09h

39 000C BA 002Er lea dx, msgpause

40 000F CD 21 int 21h

41 0011 E8 0090 call getch

42 0014 3C 2A cmp al, '\*'

43 0016 74 71 je program\_end

44 0018 3C 20 cmp al, ' '

45 001A 72 76 jb message

46 001C 3C FF cmp al, 'я'

47 001E 77 72 ja message

48 0020 cycle:

49 0020 50 push ax

50 0021 52 push dx

51 0022 E8 008F call clrscr

52 0025 B4 09 mov ah, 09h

53 0027 BA 0016r lea dx, strheader

54 002A CD 21 int 21h

55 002C E8 007A call clrf

56 002F 5A pop dx

57 0030 58 pop ax

Turbo Assembler Version 3.1 03/23/25 16:12:29 Page 2

laba4.asm

58 0031 B9 0014 mov cx, 20

59 0034 8A D8 mov bl, al

60 0036 text:

61 0036 50 push ax

62 0037 B2 BA mov dl, 'є'

63 0039 E8 0063 call putch

64 003C B2 20 mov dl, ' '

65 003E E8 005E call putch

66 0041 8A D3 mov dl, bl

67 0043 E8 0059 call putch

68 0046 B2 20 mov dl, ' '

69 0048 E8 0054 call putch

70 004B B2 3D mov dl, '='

71 004D E8 004F call putch

72 0050 B2 20 mov dl, ' '

73 0052 E8 004A call putch

74 0055 58 pop ax

75 0056 53 push bx

76 0057 E8 006C call to\_hex

77 005A 5B pop bx

78 005B B2 BA mov dl, 'є'

79 005D E8 003F call putch

80 0060 FE C3 inc bl

81 0062 E8 0044 call clrf

82 0065 8B C3 mov ax, bx

83 0067 E2 CD loop text

84 0069 B4 09 mov ah, 09h

85 006B BA 0022r lea dx, strfooter

86 006E CD 21 int 21h

87 0070 E8 0036 call clrf

88 0073 B4 09 mov ah, 09h

89 0075 BA 002Er lea dx, msgpause

90 0078 CD 21 int 21h

91 007A E8 0027 call getch

92 007D 3C 20 cmp al, ' '

93 007F 72 11 jb message

94 0081 3C FF cmp al, 'я'

95 0083 77 0D ja message

96 ; Проверяем, какой символ ввели с клавиатуры

97 0085 3C 2A cmp al, '\*'

98 ; Если ввели символ, отличный от \*, повторяем вывод символов

99 0087 75 97 jne cycle

100 ; Очистка экрана и выход

101 0089 program\_end:

102 0089 E8 0028 call clrscr

103 008C B0 05 mov al, 05h

104 008E B4 4C mov ah, 4ch

105 0090 CD 21 int 21h

106 0092 message:

107 0092 B4 09 mov ah, 09h

108 0094 BA 008Br lea dx, msgerror

109 0097 CD 21 int 21h

110 0099 E8 000D call clrf

111 009C E9 FF6B jmp after\_message

112 ; Процедура вывода символа на экран

113 009F putch proc

114 009F B4 02 mov ah, 02h

Turbo Assembler Version 3.1 03/23/25 16:12:29 Page 3

laba4.asm

115 00A1 CD 21 int 21h

116 00A3 C3 ret

117 00A4 putch endp

118 ; Процедура ожидания ввода символа с клавиатуры

119 00A4 getch proc

120 00A4 B4 08 mov ah, 08h

121 00A6 CD 21 int 21h

122 00A8 C3 ret

123 00A9 getch endp

124 ; Процедура перевода строки

125 00A9 clrf proc

126 00A9 B2 0A mov dl, 0Ah

127 00AB E8 FFF1 call putch

128 00AE B2 0D mov dl, 0Dh

129 00B0 E8 FFEC call putch

130 00B3 C3 ret

131 00B4 clrf endp

132 ; Процедура очистки экрана

133 00B4 clrscr proc

134 00B4 B4 00 mov ah, 00h

135 00B6 B0 03 mov al, 3

136 00B8 CD 10 int 10h

137 00BA B9 0019 mov cx, 25

138 00BD labclr:

139 00BD B4 02 mov ah, 02h

140 00BF B2 0A mov dl, 10

141 00C1 CD 21 int 21h

142 00C3 E2 F8 loop labclr

143 00C5 C3 ret

144 00C6 clrscr endp

145 ; Процедура перевода числа в шестнадцатиричное представление

146 00C6 to\_hex proc

147 00C6 1E push ds

148 00C7 07 pop es

149 00C8 BF 0010r lea di, string

150 00CB BB 0000r mov bx, offset hex

151 00CE BE 0000 mov si, 0

152 00D1 B4 00 mov ah, 0

153 00D3 50 push ax

154 00D4 D0 E8 D0 E8 D0 E8 D0+ shr al, 4

155 E8

156 00DC D7 xlat

157 00DD AA stosb

158 00DE 58 pop ax

159 00DF 50 push ax

160 00E0 24 0F and al, 00001111b

161 00E2 D7 xlat

162 00E3 AA stosb

163 00E4 58 pop ax

164 00E5 B8 0068 mov ax, 'h'

165 00E8 AB stosw

166 00E9 B4 09 mov ah, 09h

167 00EB BA 0010r lea dx, string

168 00EE CD 21 int 21h

169 00F0 C3 ret

170 00F1 to\_hex endp

171 00F1 mycode ends

Turbo Assembler Version 3.1 03/23/25 16:12:29 Page 4

laba4.asm

172 end start

Turbo Assembler Version 3.1 03/23/25 16:12:29 Page 5

Symbol Table

Symbol Name Type Value Cref (defined at #)

??DATE Text "03/23/25"

??FILENAME Text "laba4 "

??TIME Text "16:12:29"

??VERSION Number 030A

@CPU Text 0101H

@CURSEG Text MYCODE #2 #29

@FILENAME Text LABA4

@WORDSIZE Text 2 #2 #29

AFTER\_MESSAGE Near MYCODE:000A #37 111

CLRF Near MYCODE:00A9 55 81 87 110 #125

CLRSCR Near MYCODE:00B4 36 51 102 #133

CYCLE Near MYCODE:0020 #48 99

GETCH Near MYCODE:00A4 41 91 #119

HEX Byte MYDATA:0000 #3 150

LABCLR Near MYCODE:00BD #138 142

MESSAGE Near MYCODE:0092 45 47 93 95 #106

MSGERROR Byte MYDATA:008B #25 108

MSGPAUSE Byte MYDATA:002E #11 39 89

MSGPAUSEERROR Byte MYDATA:005A #18

PROGRAM\_END Near MYCODE:0089 43 #101

PUTCH Near MYCODE:009F 63 65 67 69 71 73 79 #113 127 129

START Near MYCODE:0002 #33 172

STRFOOTER Byte MYDATA:0022 #9 85

STRHEADER Byte MYDATA:0016 #7 53

STRING Byte MYDATA:0010 #6 149 167

TEXT Near MYCODE:0036 #60 83

TO\_HEX Near MYCODE:00C6 76 #146

Groups & Segments Bit Size Align Combine Class Cref (defined at #)

MYCODE 16 00F1 Para none CODE #29 30 30

MYDATA 16 009F Para none DATA #2 34

# Выводы по ЛР № 4

В данной лабораторной работе были получены навыки разработки и отладки программ на ЯП, знания об использовании циклов и знания о написании методов по переводу символов на Ассемблере.