





Содержимое LST-файла:

Turbo Assembler Version 1.01 10/08/21 13:05:37 Page 1

LAB3.ASM

1 ; x=3\*5 + (1982-1857)/25+497

2 0000 wow segment 'code'

3 assume cs:wow, ds:wow, ss:wow, es:wow

4 org 100h

5 0100 EB 0B 90 begin: jmp main

6 ;---------------------------------

7 0103 ???? date dw ?

8 0105 2B my\_s db '+'

9 0106 ?? T\_Th db ?

10 0107 ?? Th db ?

11 0108 ?? Hu db ?

12 0109 ?? Tens db ?

13 010A ?? Ones db ?

14 ;---------------------------------

15 ; Переменные программы

16 010B ???? T dw ?

17 010D MAIN proc near

18 ; Команды программы

19 ; x= (125/25)+(13-5)/4\*5 + 553

20

21 010D B8 007D mov ax,125

22 0110 99 cwd

23 0111 BB 0019 mov bx,25

24 0114 F7 FB idiv bx; (125/25)

25 0116 A3 010Br mov T,ax

26

27 0119 B8 000D mov ax,13

28 011C 2D 0005 sub ax,5

29 011F 99 cwd ; 13-5

30 0120 BB 0004 mov bx,4

31 0123 F7 FB idiv bx ; (13-5)/4

32 0125 BB 0005 mov bx,5

33 0128 F7 EB imul bx

34 012A 01 06 010Br add T,ax

35

36 012E B8 0229 mov ax, 553

37 0131 03 06 010Br add ax, T

38

39 0135 A3 0103r mov date,ax

40 0138 E8 003C call disp

41

42 ; -----------------------------------------

43

44 ; y= (25+15)\*(144-4)/(75-5) + 2 \* 10 \* (115-15)/100

45

46 013B B8 0019 mov ax,25

47 013E 05 000F add ax,15

48 0141 99 cwd

49 0142 BB 0090 mov bx,144

50 0145 83 EB 04 sub bx,4

51 0148 F7 EB imul bx

52 014A 99 cwd

53 014B BB 004B mov bx,75

54 014E 83 EB 05 sub bx,5

55 0151 F7 FB idiv bx

56 0153 A3 010Br mov T,ax

57

58 0156 B8 0002 mov ax, 2

59 0159 B8 000A mov ax, 10

Turbo Assembler Version 1.01 10/08/21 13:05:37 Page 2

LAB3.ASM

60 015C F7 E8 imul ax

61 015E 99 cwd

62 015F BB 0073 mov bx, 115

63 0162 83 EB 0F sub bx, 15

64 0165 F7 EB imul bx

65 0167 BB 0064 mov bx, 100

66 016A F7 FB idiv bx

67 016C 01 06 010Br add T,ax

68

69

70 0170 A3 0103r mov date,ax

71 0173 E8 0001 call disp

72 0176 C3 ret

73 0177 MAIN endp

74

75 ; Процедура выводит результат вычислений, помещенный в data

76 0177 DISP proc near

77 ;----- Вывод результата на экран ----------------

78 ;--- Число отрицательное ?----------

79 0177 A1 0103r mov ax,date

80 017A 25 8000 and ax,1000000000000000b

81 017D B1 0F mov cl,15

82 017F D3 E8 shr ax,cl

83 0181 3D 0001 cmp ax,1

84 0184 75 0D jne @m1

85 0186 A1 0103r mov ax,date

86 0189 F7 D8 neg ax

87 018B C6 06 0105r 2D mov my\_s,'-'

88 0190 EB 04 90 jmp @m2

89 ;--- Получаем десятки тысяч ---------------

90 0193 A1 0103r @m1: mov ax,date

91 0196 99 @m2: cwd

92 0197 BB 2710 mov bx,10000

93 019A F7 FB idiv bx

94 019C A2 0106r mov T\_Th,al

95 ;------- Получаем тысячи ------------------------------

96 019F 8B C2 mov ax,dx

97 01A1 99 cwd

98 01A2 BB 03E8 mov bx,1000

99 01A5 F7 FB idiv bx

100 01A7 A2 0107r mov Th,al

101 ;------ Получаем сотни ---------------

102 01AA 8B C2 mov ax,dx

103 01AC B3 64 mov bl,100

104 01AE F6 FB idiv bl

105 01B0 A2 0108r mov Hu,al

106 ;---- Получаем десятки и единицы ----------------------

107 01B3 8A C4 mov al,ah

108 01B5 98 cbw

109 01B6 B3 0A mov bl,10

110 01B8 F6 FB idiv bl

111 01BA A2 0109r mov Tens,al

112 01BD 88 26 010Ar mov Ones,ah

113 ;--- Выводим знак -----------------------

114 01C1 80 3E 0105r 2B cmp my\_s,'+'

115 01C6 74 08 je @m500

116 01C8 B4 02 mov ah,02h

117 01CA 8A 16 0105r mov dl,my\_s

118 01CE CD 21 int 21h

Turbo Assembler Version 1.01 10/08/21 13:05:37 Page 3

LAB3.ASM

119 ;---------- Выводим цифры -----------------

120 01D0 80 3E 0106r 00 @m500: cmp T\_TH,0 ; проверка на ноль

121 01D5 74 0B je @m200

122 01D7 B4 02 mov ah,02h ; выводим на экран, если не ноль

123 01D9 8A 16 0106r mov dl,T\_Th

124

125 01DD 80 C2 30 add dl,48

126 01E0 CD 21 int 21h

127 01E2 80 3E 0106r 00 @m200: cmp T\_Th,0

128 01E7 75 07 jne @m300

129 01E9 80 3E 0107r 00 cmp Th,0

130 01EE 74 0B je @m400

131 01F0 B4 02 @m300: mov ah,02h

132 01F2 8A 16 0107r mov dl,Th

133 01F6 80 C2 30 add dl,48

134 01F9 CD 21 int 21h

135 01FB 80 3E 0106r 00 @m400: cmp T\_TH,0

136 0200 75 0E jne @m600

137 0202 80 3E 0107r 00 cmp Th,0

138 0207 75 07 jne @m600

139 0209 80 3E 0108r 00 cmp hu,0

140 020E 74 0B je @m700

141 0210 B4 02 @m600: mov ah,02h

142 0212 8A 16 0108r mov dl,Hu

143 0216 80 C2 30 add dl,48

144 0219 CD 21 int 21h

145 021B 80 3E 0106r 00 @m700: cmp T\_TH,0

146 0220 75 15 jne @m900

147 0222 80 3E 0107r 00 cmp Th,0

148 0227 75 0E jne @m900

149 0229 80 3E 0108r 00 cmp Hu,0

150 022E 75 07 jne @m900

151 0230 80 3E 0109r 00 cmp Tens,0

152 0235 74 0B je @m950

153 0237 B4 02 @m900: mov ah,02h

154 0239 8A 16 0109r mov dl,Tens

155 023D 80 C2 30 add dl,48

156 0240 CD 21 int 21h

157 0242 B4 02 @m950: mov ah,02h

158 0244 8A 16 010Ar mov dl,Ones

159 0248 80 C2 30 add dl,48

160 024B CD 21 int 21h

161 024D B4 02 mov ah,02h

162 024F B2 0A mov dl,10

163 0251 CD 21 int 21h

164 0253 B4 02 mov ah,02h

165 0255 B2 0D mov dl,13

166 0257 CD 21 int 21h

167 ;-------------------------------------

168 0259 B4 08 mov ah,08

169 025B CD 21 int 21h

170 025D C3 ret

171 025E DISP endp

172 025E wow ends

173 end begin

Turbo Assembler Version 1.01 10/08/21 13:05:37 Page 4

Symbol Table

Symbol Name Type Value

??DATE Text "10/08/21"

??FILENAME Text "LAB3 "

??TIME Text "13:05:37"

??VERSION Number 0101

@CPU Text 0101H

@CURSEG Text WOW

@FILENAME Text LAB3

@M1 Near WOW:0193

@M2 Near WOW:0196

@M200 Near WOW:01E2

@M300 Near WOW:01F0

@M400 Near WOW:01FB

@M500 Near WOW:01D0

@M600 Near WOW:0210

@M700 Near WOW:021B

@M900 Near WOW:0237

@M950 Near WOW:0242

@WORDSIZE Text 2

BEGIN Near WOW:0100

DATE Word WOW:0103

DISP Near WOW:0177

HU Byte WOW:0108

MAIN Near WOW:010D

MY\_S Byte WOW:0105

ONES Byte WOW:010A

T Word WOW:010B

TENS Byte WOW:0109

TH Byte WOW:0107

T\_TH Byte WOW:0106

Groups & Segments Bit Size Align Combine Class

WOW 16 025E Para none CODE

**Содержимое ASM-файла:**

; x=3\*5 + (1982-1857)/25+497

wow segment 'code'

assume cs:wow, ds:wow, ss:wow, es:wow

org 100h

begin: jmp main

;---------------------------------

date dw ?

my\_s db '+'

T\_Th db ?

Th db ?

Hu db ?

Tens db ?

Ones db ?

;---------------------------------

; Переменные программы

T dw ?

MAIN proc near

; Команды программы

; x= (125/25)+(13-5)/4\*5 + 553

mov ax,125

cwd

mov bx,25

idiv bx; (125/25)

mov T,ax

mov ax,13

sub ax,5

cwd ; 13-5

mov bx,4

idiv bx ; (13-5)/4

mov bx,5

imul bx

add T,ax

mov ax, 553

add ax, T

mov date,ax

call disp

; -----------------------------------------

; y= (25+15)\*(144-4)/(75-5) + 2 \* 10 \* (115-15)/100

mov ax,25

add ax,15

cwd

mov bx,144

sub bx,4

imul bx

cwd

mov bx,75

sub bx,5

idiv bx

mov T,ax

mov ax, 2

mov ax, 10

imul ax

cwd

mov bx, 115

sub bx, 15

imul bx

mov bx, 100

idiv bx

add T,ax

mov date,ax

call disp

ret

MAIN endp

; Процедура выводит результат вычислений, помещенный в data

DISP proc near

;----- Вывод результата на экран ----------------

;--- Число отрицательное ?----------

mov ax,date

and ax,1000000000000000b

mov cl,15

shr ax,cl

cmp ax,1

jne @m1

mov ax,date

neg ax

mov my\_s,'-'

jmp @m2

;--- Получаем десятки тысяч ---------------

@m1: mov ax,date

@m2: cwd

mov bx,10000

idiv bx

mov T\_Th,al

;------- Получаем тысячи ------------------------------

mov ax,dx

cwd

mov bx,1000

idiv bx

mov Th,al

;------ Получаем сотни ---------------

mov ax,dx

mov bl,100

idiv bl

mov Hu,al

;---- Получаем десятки и единицы ----------------------

mov al,ah

cbw

mov bl,10

idiv bl

mov Tens,al

mov Ones,ah

;--- Выводим знак -----------------------

cmp my\_s,'+'

je @m500

mov ah,02h

mov dl,my\_s

int 21h

;---------- Выводим цифры -----------------

@m500: cmp T\_TH,0 ; проверка на ноль

je @m200

mov ah,02h ; выводим на экран, если не ноль

mov dl,T\_Th

add dl,48

int 21h

@m200: cmp T\_Th,0

jne @m300

cmp Th,0

je @m400

@m300: mov ah,02h

mov dl,Th

add dl,48

int 21h

@m400: cmp T\_TH,0

jne @m600

cmp Th,0

jne @m600

cmp hu,0

je @m700

@m600: mov ah,02h

mov dl,Hu

add dl,48

int 21h

@m700: cmp T\_TH,0

jne @m900

cmp Th,0

jne @m900

cmp Hu,0

jne @m900

cmp Tens,0

je @m950

@m900: mov ah,02h

mov dl,Tens

add dl,48

int 21h

@m950: mov ah,02h

mov dl,Ones

add dl,48

int 21h

mov ah,02h

mov dl,10

int 21h

mov ah,02h

mov dl,13

int 21h

;-------------------------------------

mov ah,08

int 21h

ret

DISP endp

wow ends

end begin

Работа программы:

