­­­­­НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт до комп’ютернего практикуму з дисципліни

«Системне програмне забезпечення»

Прийняв

асистент кафедри ІПІ

Пархоменко А.В.

“21” травня 2023 р.

Виконав

Студент групи ІП-15

Мєшков А. І.

Київ – 2023

**Комп‘ютерний практикум № 2**

**Засоби обміну даними**

**Загальні положення**

Викладені в лекційному матеріалі.

**Завдання комп’ютерного практикуму №2**

1. Написати програму з використанням 2-х процедур:
   1. Процедура введення і перетворення цілого числа. Після цього треба виконати математичну дію над числом (-8).
   2. Процедура переведення отриманого результату в рядок та виведення його на екран.
2. Програма повинна мати захист від некоректного введення вхідних даних (символи, переповнення, ділення на 0 і т.і.)

**Текст програми**

STREG SEGMENT PARA STACK "STACK"

DW 64 DUP ('?')

STREG ENDS

DSEG SEGMENT PARA PUBLIC "DATA"

len dw 0

messtr db "Enter number -> $"

numstr db 7,?,7 dup('?')

num dw 0

fl db 0

error\_mes db "Error$"

DSEG ENDS

CSEG SEGMENT PARA PUBLIC "CODE"

ASSUME ds:DSEG, cs:CSEG, ss:STREG

READ PROC FAR

start:

PUSH ds

MOV ax, 0

PUSH ax

MOV ax,DSEG

MOV ds, ax

MOV num, 0

MOV fl, 0

MOV len, 0

;print message

MOV dx,offset messtr

MOV ah, 9

int 21h

;read number

LEA dx,numstr

MOV ah,10

int 21h

MOV al,10

int 29h

MOV ax,0

MOV al, [numstr+1]

MOV len,ax

MOV si,2

MOV al,numstr[si]

;translate to print

CMP al,2Dh

jne no\_minus

inc si

MOV fl,1

dec len

no\_minus:

MOV cx, len

MOV bx,10

cycle:

MOV ax,num

IMUL bx

jo error

MOV num,ax

MOV ax,0

MOV al,numstr[si]

SUB al,30h

CMP al,0

jl error

CMP al,9

ja error

ADD num,ax

CMP num,32768

ja error

inc si

loop cycle

CMP fl,1

jne finish

CMP num,32760

ja error

NEG num

jmp finish

error:

MOV dx,offset error\_mes

MOV ah,9

int 21h

RET

finish:

READ ENDP

WRITE PROC FAR

MOV bx,num

SUB bx,8

OR bx,bx

jns m1

MOV al,'-'

int 29h

neg bx

m1:

MOV ax,bx

XOR cx,cx

MOV bx,10

m2:

XOR dx,dx

DIV bx

ADD dl,'0'

PUSH dx

inc cx

TEST ax,ax

jnz m2

m3:

POP ax

int 29h

loop m3

RET

WRITE ENDP

MAIN PROC FAR

CALL READ

CALL WRITE

RET

MAIN ENDP

CSEG ENDS

END MAIN

**Введені та отримані результати**

Вміст .lst файлу

Turbo Assembler Version 4.0 03/27/23 12:26:25 Page 1

kp2.asm

1 0000 STREG SEGMENT PARA STACK "STACK"

2 0000 40\*(003F) DW 64 DUP ('?')

3 0080 STREG ENDS

4

5 0000 DSEG SEGMENT PARA PUBLIC "DATA"

6 0000 0000 len dw 0

7 0002 45 6E 74 65 72 20 6E+ messtr db "Enter number -> $"

8 75 6D 62 65 72 20 2D+

9 3E 20 24

10 0013 07 ?? 07\*(3F) numstr db 7,?,7 dup('?')

11 001C 0000 num dw 0

12 001E 00 fl db 0

13 001F 45 72 72 6F 72 24 error\_mes db "Error$"

14 0025 DSEG ENDS

15

16 0000 CSEG SEGMENT PARA PUBLIC "CODE"

17 ASSUME ds:DSEG, cs:CSEG, ss:STREG

18

19 0000 READ PROC FAR

20

21 0000 start:

22 0000 1E PUSH ds

23 0001 B8 0000 MOV ax, 0

24 0004 50 PUSH ax

25 0005 B8 0000s MOV ax,DSEG

26 0008 8E D8 MOV ds, ax

27

28 000A C7 06 001Cr 0000 MOV num, 0

29 0010 C6 06 001Er 00 MOV fl, 0

30 0015 C7 06 0000r 0000 MOV len, 0

31

32 ;print message

33 001B BA 0002r MOV dx,offset messtr

34 001E B4 09 MOV ah, 9

35 0020 CD 21 int 21h

36

37 ;read number

38 0022 BA 0013r LEA dx,numstr

39 0025 B4 0A MOV ah,10

40 0027 CD 21 int 21h

41

42 0029 B0 0A MOV al,10

43 002B CD 29 int 29h

44

45 002D B8 0000 MOV ax,0

46 0030 A0 0014r MOV al, [numstr+1]

47 0033 A3 0000r MOV len,ax

48

49 0036 BE 0002 MOV si,2

50 0039 8A 84 0013r MOV al,numstr[si]

51

52 ;translate to print

53 003D 3C 2D CMP al,2Dh

54 003F 75 0A jne no\_minus

55

56 0041 46 inc si

57 0042 C6 06 001Er 01 MOV fl,1

Turbo Assembler Version 4.0 03/27/23 12:26:25 Page 2

kp2.asm

58 0047 FF 0E 0000r dec len

59

60 004B no\_minus:

61 004B 8B 0E 0000r MOV cx, len

62 004F BB 000A MOV bx,10

63

64 0052 cycle:

65 0052 A1 001Cr MOV ax,num

66 0055 F7 EB IMUL bx

67 0057 70 39 jo error

68 0059 A3 001Cr MOV num,ax

69 005C B8 0000 MOV ax,0

70 005F 8A 84 0013r MOV al,numstr[si]

71 0063 2C 30 SUB al,30h

72 0065 3C 00 CMP al,0

73 0067 7C 29 jl error

74 0069 3C 09 CMP al,9

75 006B 77 25 ja error

76 006D 01 06 001Cr ADD num,ax

77 0071 81 3E 001Cr 8000 CMP num,32768

78 0077 77 19 ja error

79 0079 46 inc si

80 007A E2 D6 loop cycle

81

82 007C 80 3E 001Er 01 CMP fl,1

83 0081 75 17 jne finish

84 0083 81 3E 001Cr 7FBB CMP num,32699

85 0089 77 07 ja error

86 008B F7 1E 001Cr NEG num

87 008F EB 09 90 jmp finish

88

89 0092 error:

90 0092 BA 001Fr MOV dx,offset error\_mes

91 0095 B4 09 MOV ah,9

92 0097 CD 21 int 21h

93 0099 CB RET

94 009A finish:

95 009A READ ENDP

96

97 009A WRITE PROC FAR

98

99 009A 8B 1E 001Cr MOV bx,num

100 009E 83 EB 08 SUB bx,8

101 00A1 0B DB OR bx,bx

102 00A3 79 06 jns m1

103 00A5 B0 2D MOV al,'-'

104 00A7 CD 29 int 29h

105 00A9 F7 DB neg bx

106 00AB m1:

107 00AB 8B C3 MOV ax,bx

108 00AD 33 C9 XOR cx,cx

109 00AF BB 000A MOV bx,10

110 00B2 m2:

111 00B2 33 D2 XOR dx,dx

112 00B4 F7 F3 DIV bx

113 00B6 80 C2 30 ADD dl,'0'

114 00B9 52 PUSH dx

Turbo Assembler Version 4.0 03/27/23 12:26:25 Page 3

kp2.asm

115 00BA 41 inc cx

116 00BB 85 C0 TEST ax,ax

117 00BD 75 F3 jnz m2

118 00BF m3:

119 00BF 58 POP ax

120 00C0 CD 29 int 29h

121 00C2 E2 FB loop m3

122 00C4 CB RET

123

124 00C5 WRITE ENDP

125

126 00C5 MAIN PROC FAR

127 00C5 0E E8 FF37 CALL READ

128 00C9 0E E8 FFCD CALL WRITE

129 00CD CB RET

130 00CE MAIN ENDP

131

132 00CE CSEG ENDS

133

134 END MAIN

Turbo Assembler Version 4.0 03/27/23 12:26:25 Page 4

Symbol Table

Symbol Name Type Value

??DATE Text "03/27/23"

??FILENAME Text "kp2 "

??TIME Text "12:26:25"

??VERSION Number 0400

@CPU Text 0101H

@CURSEG Text CSEG

@FILENAME Text KP2

@WORDSIZE Text 2

CYCLE Near CSEG:0052

ERROR Near CSEG:0092

ERROR\_MES Byte DSEG:001F

FINISH Near CSEG:009A

FL Byte DSEG:001E

LEN Word DSEG:0000

M1 Near CSEG:00AB

M2 Near CSEG:00B2

M3 Near CSEG:00BF

MAIN Far CSEG:00C5

MESSTR Byte DSEG:0002

NO\_MINUS Near CSEG:004B

NUM Word DSEG:001C

NUMSTR Byte DSEG:0013

READ Far CSEG:0000

START Near CSEG:0000

WRITE Far CSEG:009A

Groups & Segments Bit Size Align Combine Class

CSEG 16 00CE Para Public CODE

DSEG 16 0025 Para Public DATA

STREG 16 0080 Para Stack STACK

Вміст .map файлу

Start Stop Length Name Class

00000H 0007FH 00080H STREG STACK

00080H 000A4H 00025H DSEG DATA

000B0H 0017DH 000CEH CSEG CODE

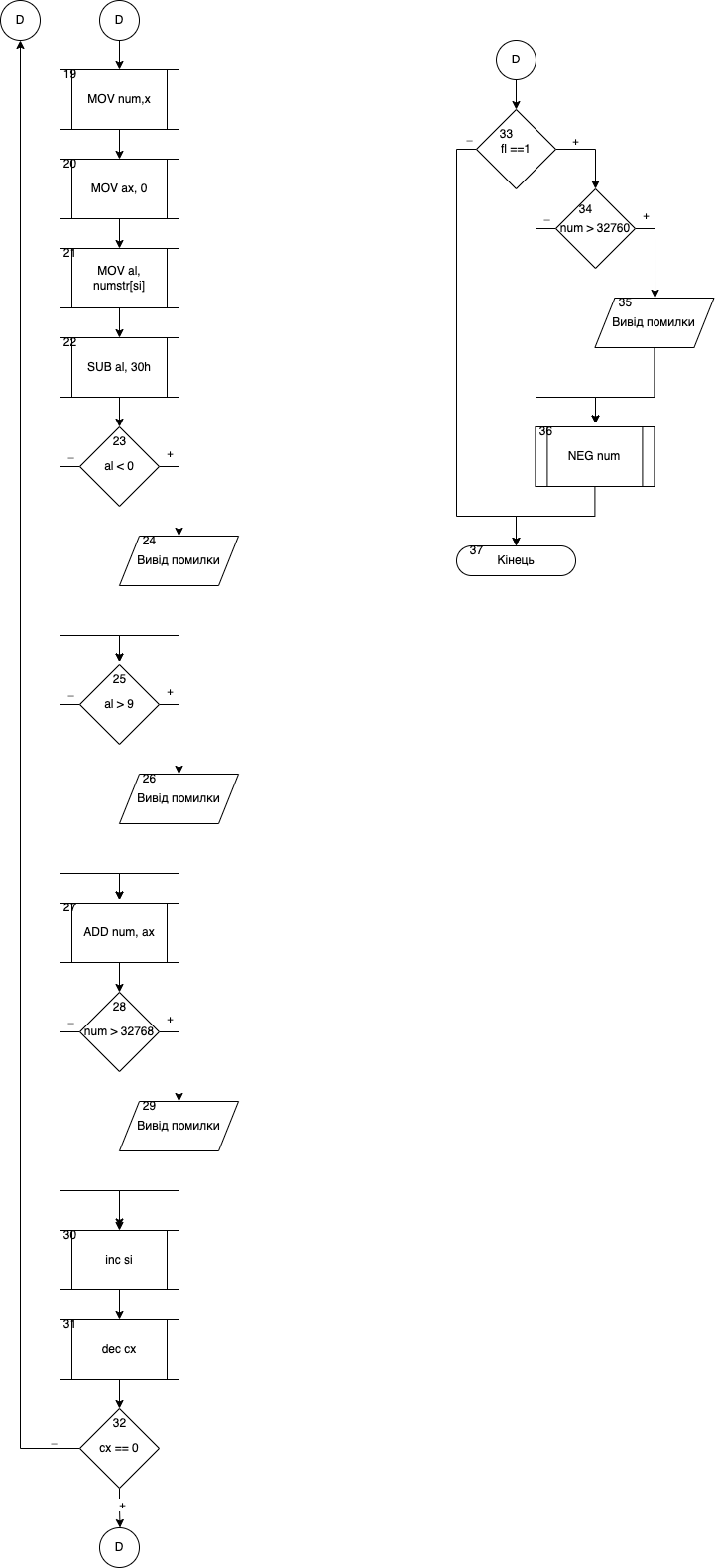
Address Publics by Name

Address Publics by Value

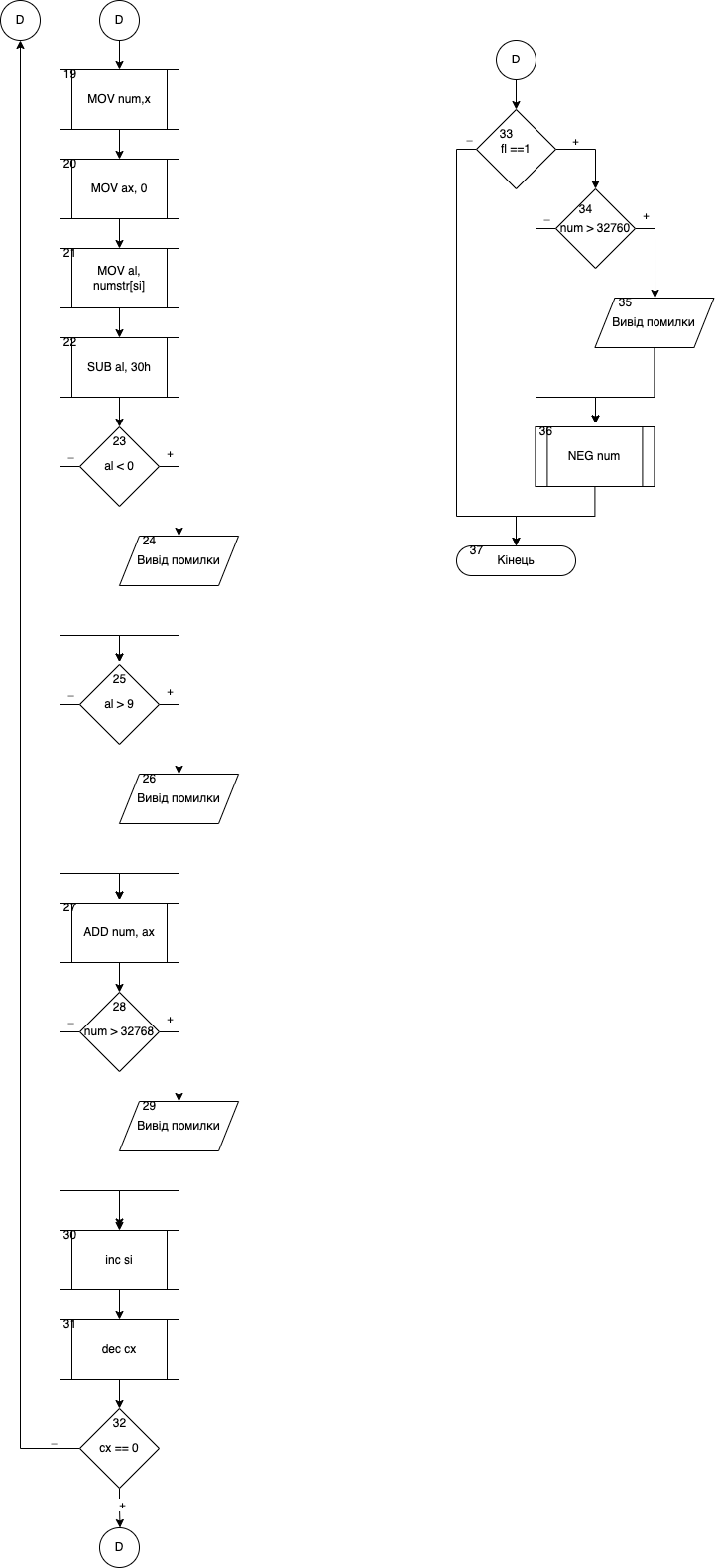
Program entry point at 000B:00C5

Схема функціонування програми

|  |
| --- |
| 5  4  3  2  1 |
| Рис. 2.1 Схема функціонування головної процедури    A  1  2  3  4  5  6  7  8  Рис. 2.2 Схема функціонування процедури читання |
| Вивід помилку  18  17  16  15  14  9  10  11  12  13  Продовження рис. 2.2 |

****

Продовження рис. 2.2

****

Продовження рис. 2.2

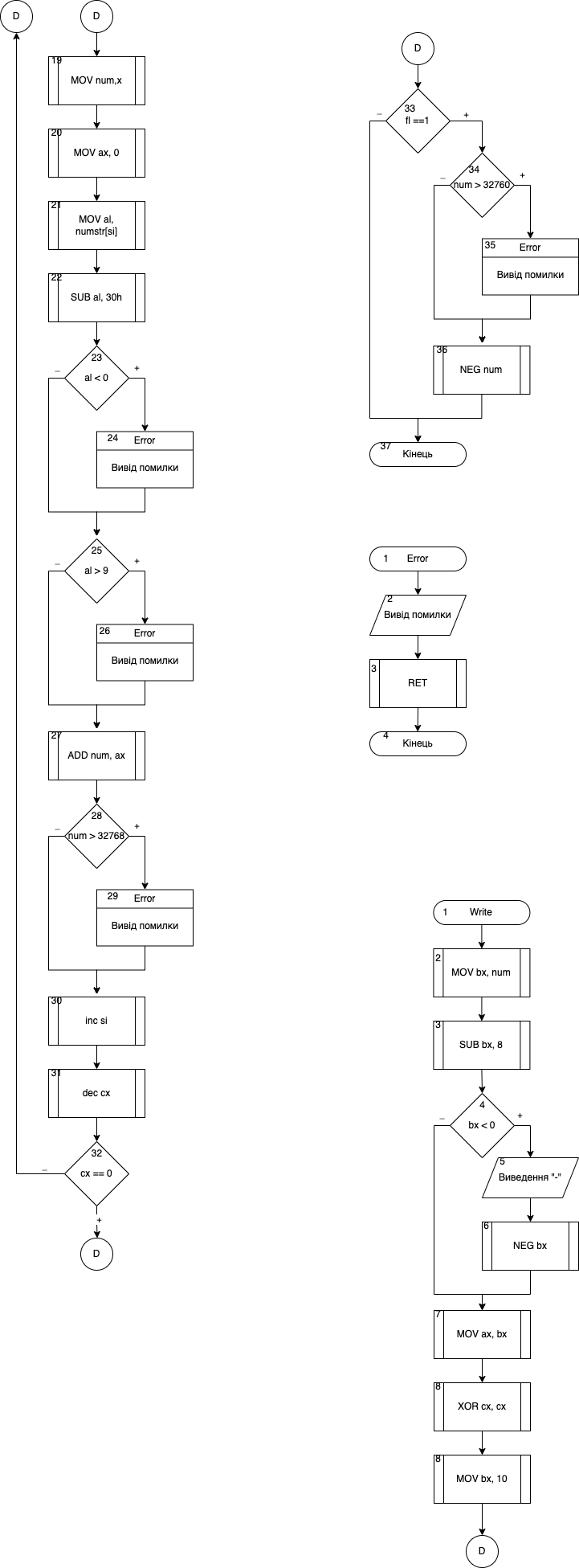
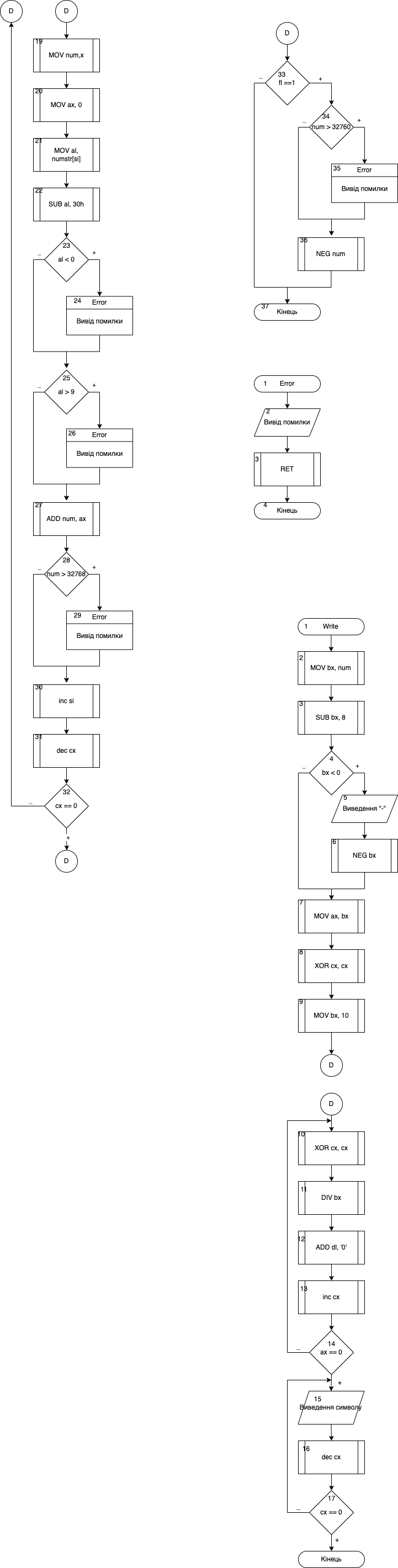
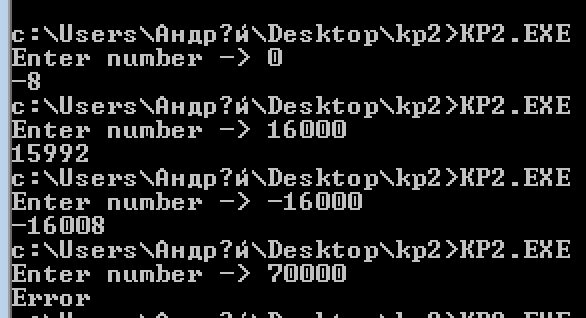
****

Рис. 2.4 Схема функціонування процедури виведення

****

Продовження рис. 2.4

**Приклади виконання програми**

****

**Висновок:** Під час виконання комп’ютерного практикуму мною було опрацьовано теоретичний матеріал по засобам вводу та виводу у мові програмування асемблер, створено програму, яка читає рядок, переводить його у число, віднімає від нього 8, перетворює число у рядок та виводить його. Було протестовано програму. В результаті тестування було з’ясовано, що програма працює для чисел в діапазоні [-32761;32768].