

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ

**Лабораторна робота № 2**

з дисципліни

**«Системне програмування»**

Виконав: студент ІІI курсу

ФІОТ групи \_\_\_\_\_

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

Перевірив:\_\_\_\_\_\_\_\_\_\_\_\_\_

\_

Київ – 2021

**Постановка задачі**

1. Написати процедуру введення і перетворення цілого числа.

2. Виконати математичну дію над числом (номер завдання вибирати за останніми двома

числами номеру в заліковій книжці).

3. Перевести число в рядок та вивести його на екран.

**Варіант 16:** \*7

**Результат**

**Створений asm файл**

;; main.asm

DATA\_SEG SEGMENT PARA PUBLIC "DATA"

hello\_string db 14, "Input number: "

input\_digit db 5, ?, 5 dup("?")

error\_msg db 12, "Invalid data"

DATA\_SEG ENDS

CODE\_SEG SEGMENT PARA PUBLIC "CODE"

ASSUME CS:CODE\_SEG, DS:DATA\_SEG

OUTPUT\_PROC PROC NEAR

mov ah, 02h

printing:

mov dl, [di]

int 21h

inc di

loop printing

ret

OUTPUT\_PROC ENDP

INPUT\_VALUE PROC NEA

mov ah, 10

int 21h

mov dl,0ah

mov ah,02

int 21h

ret

INPUT\_VALUE ENDP

STR\_TO\_INT PROC NEAR

cmp byte ptr [di], '-'

jne positiv\_num

mov si, 1

inc di

positiv\_num:

xor ax, ax

mov bx, 10

main\_array\_cycle:

mov cl, [di]

cmp cl, 0Dh

je end\_line

cmp cl, '0'

jl error

cmp cl, '9'

jg error

sub cl, '0'

mul bx ; ax\*bx -> (dx, ax)

add ax, cx

inc di

jmp main\_array\_cycle

error:

lea di, error\_msg[1]

mov cx, 0

mov cl, error\_msg[0]

call OUTPUT\_PROC

mov ax, 4c00h

int 21h

end\_line:

cmp si, 1

jne not\_negative

neg ax

not\_negative:

ret

STR\_TO\_INT ENDP

INT\_TO\_STR PROC NEAR

mov bx, ax

or bx, bx

jns positive\_value

mov al, '-'

int 29h

neg bx

positive\_value:

mov ax, bx

xor cx, cx

mov bx, 10

main\_calc\_cycle:

xor dx, dx

div bx

add dl, '0'

push dx

inc cx

cmp ax, 0

jnz main\_calc\_cycle

print:

pop ax

int 29h

loop print

ret

INT\_TO\_STR ENDP

; //////////////////////{ MAIN }/////////////////////

MAIN PROC FAR

push DS

xor ax, ax

push ax

mov ax, DATA\_SEG

mov DS, ax

lea di, hello\_string[1]

mov cx, 0

mov cl, hello\_string[0]

call OUTPUT\_PROC

lea dx, input\_digit

call INPUT\_VALUE

lea di, input\_digit[2]

call STR\_TO\_INT

mov bx, 7

mul bx

call INT\_TO\_STR

ret

MAIN ENDP

CODE\_SEG ENDS

END MAIN

**Лістинг программи**

Turbo Assembler Version 4.0 02/01/21 16:10:05 Page 1

lab2\main.asm

1 ;; main.asm

2

3 0000 DATA\_SEG SEGMENT PARA PUBLIC "DATA"

4 0000 0E 49 6E 70 75 74 20+ hello\_string db 14, "Input number: "

5 6E 75 6D 62 65 72 3A+

6 20

7 000F 05 ?? 05\*(3F) input\_digit db 5, ?, 5 dup("?")

8 0016 0C 49 6E 76 61 6C 69+ error\_msg db 12, "Invalid data"

9 64 20 64 61 74 61

10 0023 DATA\_SEG ENDS

11

12

13 0000 CODE\_SEG SEGMENT PARA PUBLIC "CODE"

14

15 ASSUME CS:CODE\_SEG, DS:DATA\_SEG

16

17 0000 OUTPUT\_PROC PROC NEAR

18

19 0000 B4 02 mov ah, 02h

20 0002 printing:

21 0002 8A 15 mov dl, [di]

22 0004 CD 21 int 21h

23 0006 47 inc di

24 0007 E2 F9 loop printing

25

26 0009 C3 ret

27 000A OUTPUT\_PROC ENDP

28

29

30 000A INPUT\_VALUE PROC NEA

31

32 000A B4 0A mov ah, 10

33 000C CD 21 int 21h

34

35 000E B2 0A mov dl,0ah

36 0010 B4 02 mov ah,02

37 0012 CD 21 int 21h

38

39 0014 C3 ret

40 0015 INPUT\_VALUE ENDP

41

42

43 0015 STR\_TO\_INT PROC NEAR

44

45 0015 80 3D 2D cmp byte ptr [di], '-'

46 0018 75 04 jne positiv\_num

47 001A BE 0001 mov si, 1

48 001D 47 inc di

49 001E positiv\_num:

50 001E 33 C0 xor ax, ax

51 0020 BB 000A mov bx, 10

52 0023 main\_array\_cycle:

53

54 0023 8A 0D mov cl, [di]

55 0025 80 F9 0D cmp cl, 0Dh

56 0028 74 26 je end\_line

57

\_Turbo Assembler Version 4.0 02/01/21 16:10:05 Page 2

lab2\main.asm

58 002A 80 F9 30 cmp cl, '0'

59 002D 7C 0F jl error

60 002F 80 F9 39 cmp cl, '9'

61 0032 7F 0A jg error

62

63 0034 80 E9 30 sub cl, '0'

64 0037 F7 E3 mul bx ; ax\*bx -> (dx, ax)

65 0039 03 C1 add ax, cx

66 003B 47 inc di

67 003C EB E5 jmp main\_array\_cycle

68

69 003E error:

70 003E BF 0017r lea di, error\_msg[1]

71 0041 B9 0000 mov cx, 0

72 0044 8A 0E 0016r mov cl, error\_msg[0]

73 0048 E8 FFB5 call OUTPUT\_PROC

74 004B B8 4C00 mov ax, 4c00h

75 004E CD 21 int 21h

76

77 0050 end\_line:

78 0050 83 FE 01 cmp si, 1

79 0053 75 02 jne not\_negative

80 0055 F7 D8 neg ax

81 0057 not\_negative:

82

83 0057 C3 ret

84 0058 STR\_TO\_INT ENDP

85

86

87 0058 INT\_TO\_STR PROC NEAR

88

89 0058 8B D8 mov bx, ax

90

91 005A 0B DB or bx, bx

92 005C 79 06 jns positive\_value

93 005E B0 2D mov al, '-'

94 0060 CD 29 int 29h

95 0062 F7 DB neg bx

96 0064 positive\_value:

97 0064 8B C3 mov ax, bx

98 0066 33 C9 xor cx, cx

99 0068 BB 000A mov bx, 10

100

101 006B main\_calc\_cycle:

102 006B 33 D2 xor dx, dx

103 006D F7 F3 div bx

104 006F 80 C2 30 add dl, '0'

105 0072 52 push dx

106 0073 41 inc cx

107 0074 3D 0000 cmp ax, 0 ; if zf = 0

108 0077 75 F2 jnz main\_calc\_cycle

109

110 0079 print:

111 0079 58 pop ax

112 007A CD 29 int 29h

113 007C E2 FB loop print

114

\_Turbo Assembler Version 4.0 02/01/21 16:10:05 Page 3

lab2\main.asm

115 007E C3 ret

116 007F INT\_TO\_STR ENDP

117

118 ; //////////////////////{ MAIN }/////////////////////

119

120 007F MAIN PROC FAR

121

122 007F 1E push DS

123 0080 33 C0 xor ax, ax

124 0082 50 push ax

125

126 0083 B8 0000s mov ax, DATA\_SEG

127 0086 8E D8 mov DS, ax

128

129

130 0088 BF 0001r lea di, hello\_string[1]

131 008B B9 0000 mov cx, 0

132 008E 8A 0E 0000r mov cl, hello\_string[0]

133 0092 E8 FF6B call OUTPUT\_PROC

134

135 0095 BA 000Fr lea dx, input\_digit

136 0098 E8 FF6F call INPUT\_VALUE

137

138 009B BF 0011r lea di, input\_digit[2]

139 009E E8 FF74 call STR\_TO\_INT

140

141 00A1 BB 0007 mov bx, 7

142 00A4 F7 E3 mul bx

143

144 00A6 E8 FFAF call INT\_TO\_STR

145

146 00A9 CB ret

147 00AA MAIN ENDP

148

149 00AA CODE\_SEG ENDS

150 END MAIN

\_Turbo Assembler Version 4.0 02/01/21 16:10:05 Page 4

Symbol Table

Symbol Name Type Value

??DATE Text "02/01/21"

??FILENAME Text "main "

??TIME Text "16:10:05"

??VERSION Number 0400

@CPU Text 0101H

@CURSEG Text CODE\_SEG

@FILENAME Text MAIN

@WORDSIZE Text 2

END\_LINE Near CODE\_SEG:0050

ERROR Near CODE\_SEG:003E

ERROR\_MSG Byte DATA\_SEG:0016

HELLO\_STRING Byte DATA\_SEG:0000

INPUT\_DIGIT Byte DATA\_SEG:000F

INPUT\_VALUE Near CODE\_SEG:000A

INT\_TO\_STR Near CODE\_SEG:0058

MAIN Far CODE\_SEG:007F

MAIN\_ARRAY\_CYCLE Near CODE\_SEG:0023

MAIN\_CALC\_CYCLE Near CODE\_SEG:006B

NEA Number [BP+0004]

NOT\_NEGATIVE Near CODE\_SEG:0057

OUTPUT\_PROC Near CODE\_SEG:0000

POSITIVE\_VALUE Near CODE\_SEG:0064

POSITIV\_NUM Near CODE\_SEG:001E

PRINT Near CODE\_SEG:0079

PRINTING Near CODE\_SEG:0002

STR\_TO\_INT Near CODE\_SEG:0015

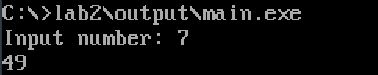
Groups & Segments Bit Size Align Combine Class

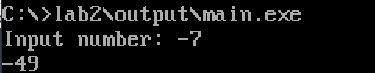
CODE\_SEG 16 00AA Para Public CODE

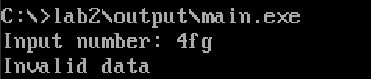
DATA\_SEG 16 0023 Para Public DATA

\_

**Результат роботи программи**

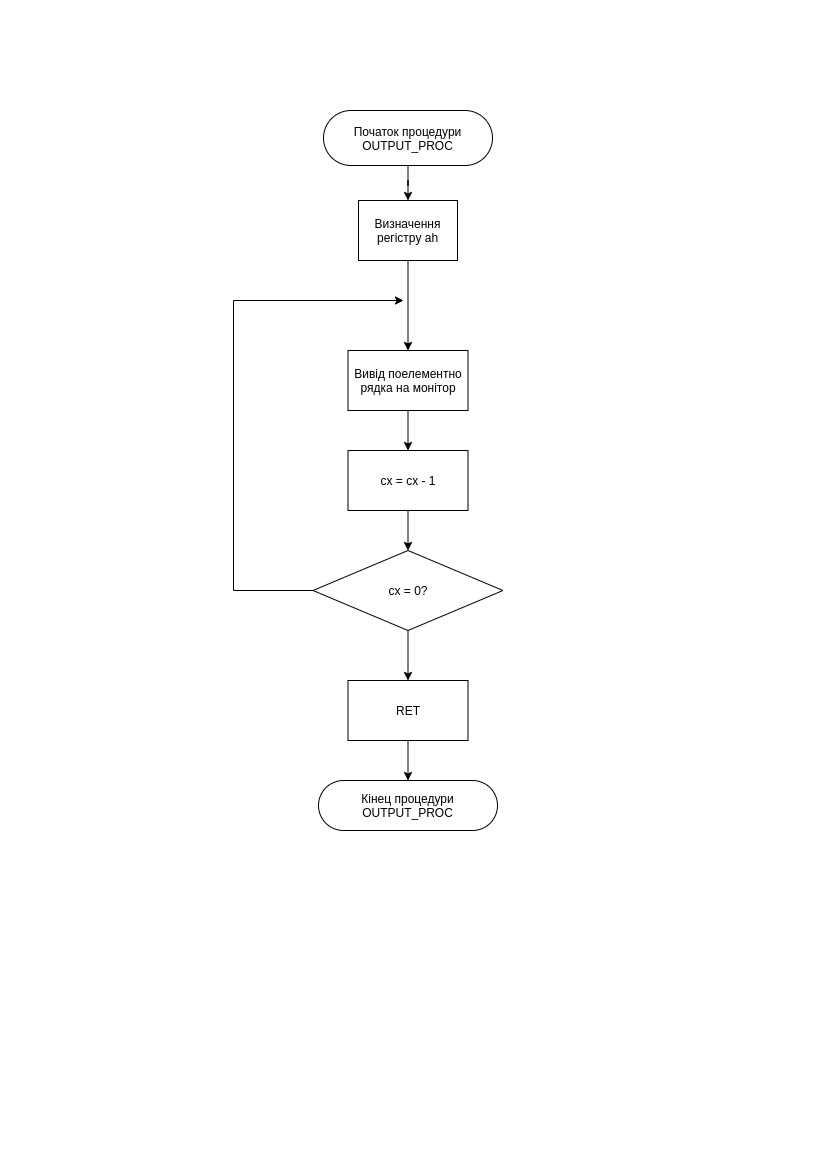
****

****

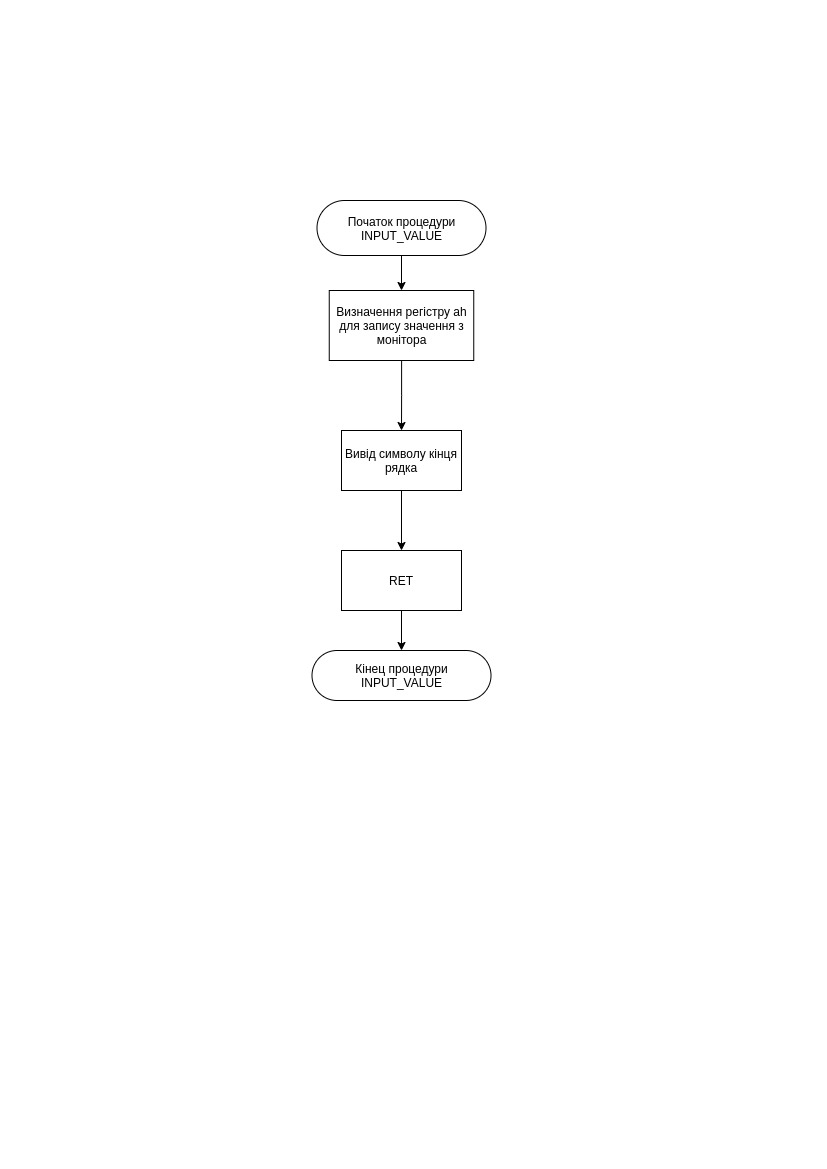
****

**Схема роботи програми**

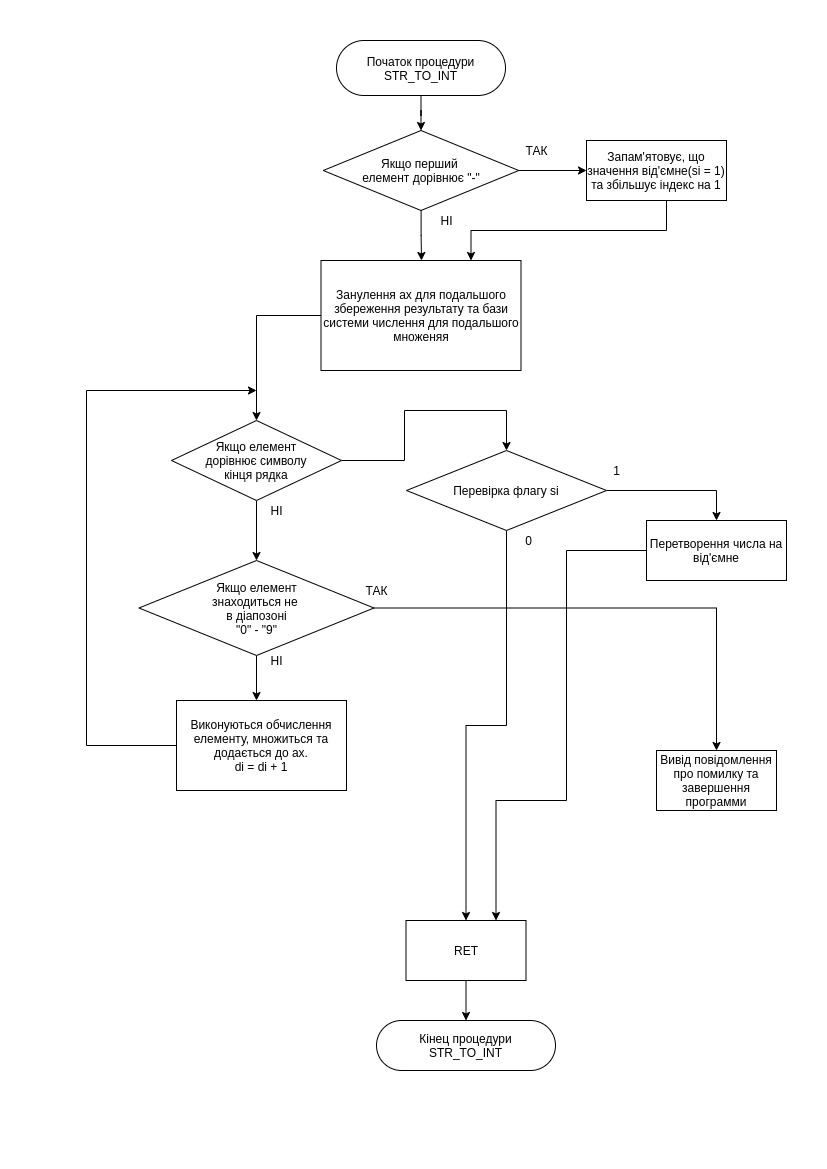
**Схема роботи процедури OUTPUT\_PROC**

****

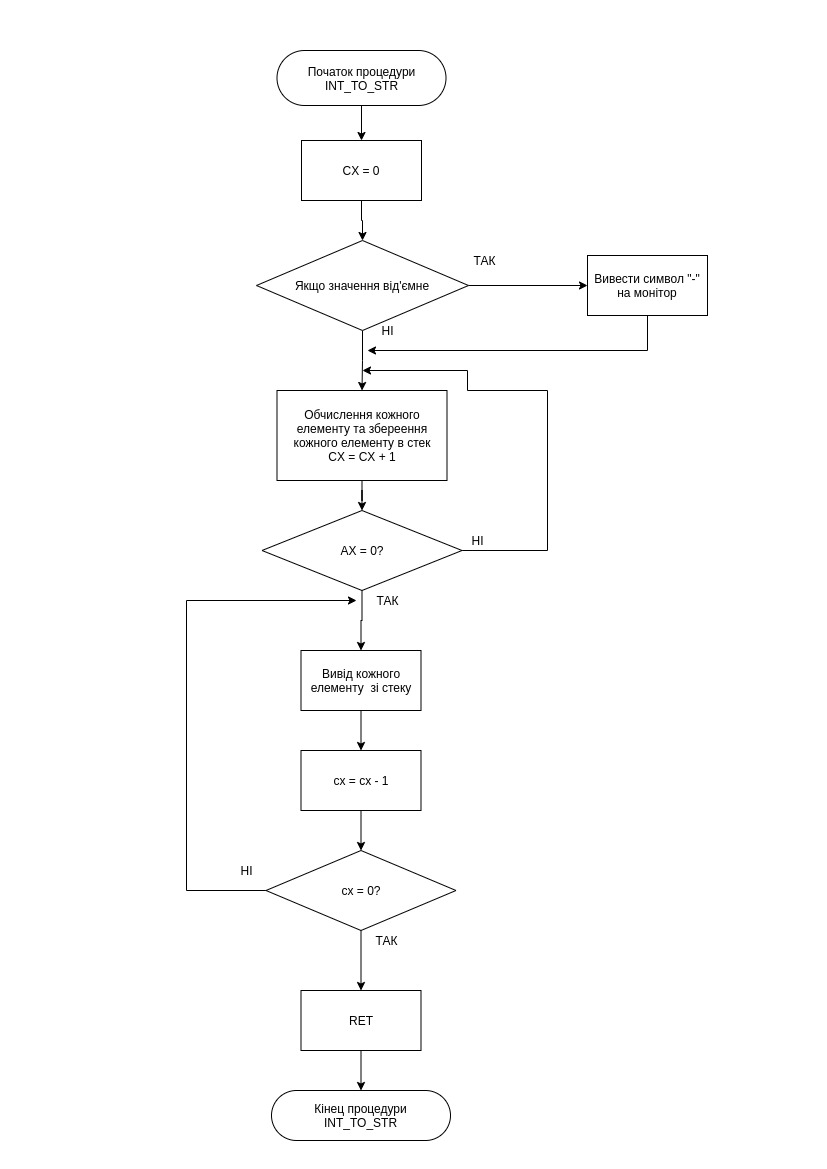
**Схема роботи процедури INPUT\_VALUE**

****

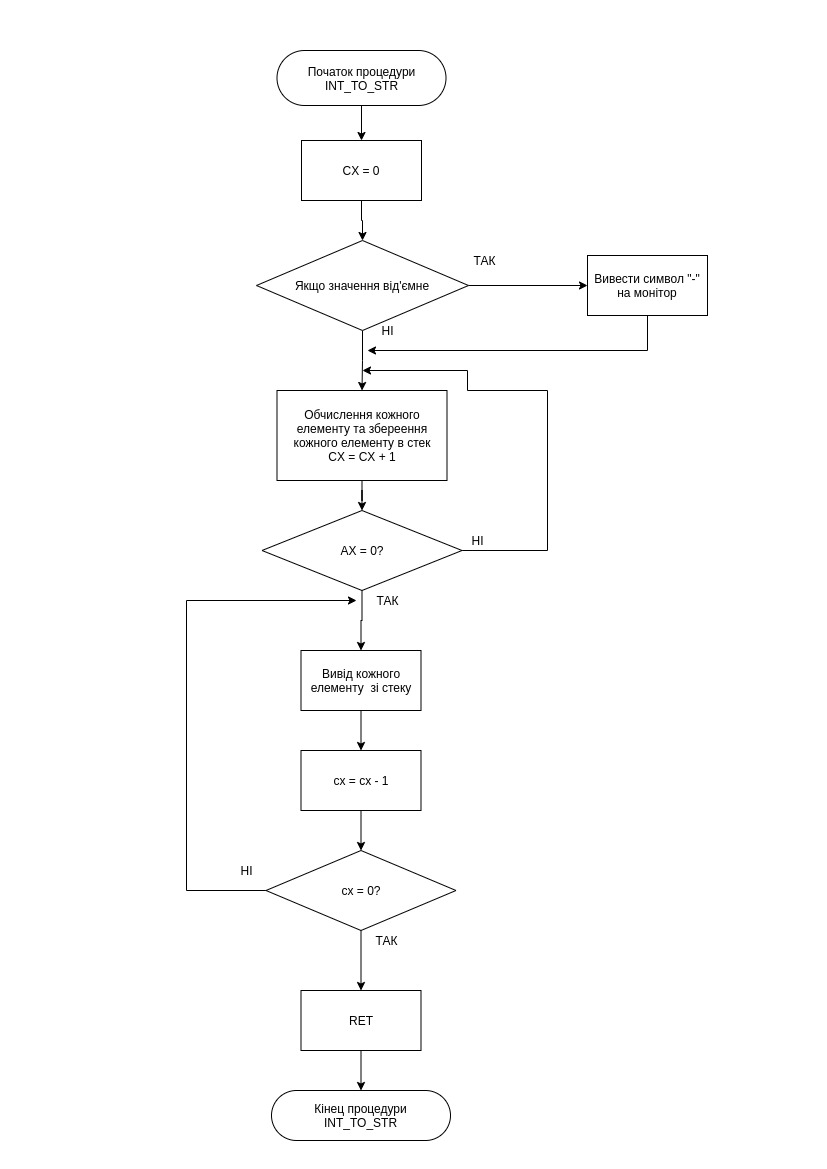
**Схема роботи процедури STR\_TO\_INT**

****

**Схема роботи процедури INT\_TO\_STR**

****

**Схема роботи процедури MAIN**

****

**Висновок**

1. Написано процедуру введення і перетворення цілого числа.

2. Виконано математичну дію над числом (номер завдання вибирати за останніми двома

числами номеру в заліковій книжці).

3. В програмі число переводиться в рядок та виводиться на екран.