****

**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський Політехнічний Інститут»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра технічної кібернетики**

**Звіт до комп‘ютерного практикуму №2 з модуля**

**“ Системне програмування ”**

Прийняв Виконав

доцент кафедри ТК Студент групи ІК-61

Лісовиченко О.І. Юдов А.М.

“...” ............... 2017 р.

**Київ 2017**

**Комп’ютерний практикум №2**

**Тема**: Засоби обміну даними.

**Завдання**:

1. Скласти процедуру введення і перетворення цілого числа.

2. Скласти і реалізувати програму введення та виведення цілого числа зі знаком та виведення рядка символів.

3. Введення та виведення цілого числа з запрошенням до користувача.

**Текст програми:**

STSEG SEGMENT PARA STACK "STACK" ;объявление сегмента стека. stseg - имя сегмента, segment - зарезервированное слово, para - тип выравнивания, кратное 16, stack - необходимо для обэъединения, "stack" - слово, по которому будет выравниваться

DB 64 DUP ("STACK") ;определение 64 слов "stack"

STSEG ENDS ;конец объявления сегмента стека

dseg segment para public 'data' ;объявление сегмента данных.

START\_STR DB "ENTER NUMBER (-999 TO 9999): ", 13, 10, '$' ;объявление строки, 13 - конец строки, 10 – возврат каретки

ERROR\_STR DB "INCORRECT INPUT", 13, 10, '$' ;объявление строки

OUTPUT\_STR DB "ENTERED NUMBER: ", 13, 10, '$' ;объявление строки

DUMP DB 5, ?, 5 DUP("$") ;запись в dump 5, ?, и 5 раз ?

DIGIT DW 0 ;объявление значения равно 0, DW - 2 байта

NEGATIVE DB 0 ;объявление знака равно 0, DB - 1 байт

DSEG ENDS ;конец объявление сегмета данных

CSEG SEGMENT PARA PUBLIC "CODE" ;начало объявления сегмента кода

MAIN PROC FAR ;начало выполнения процедуры "main", far - определяет дальность возврата

ASSUME CS:CSEG,DS:DSEG,SS:STSEG ;привязка сегментов к регистрам

PUSH DS ;помещение DS в стек

PUSH AX ;помещение AX в стек

MOV AX, 0 ;помещение 0 в регистр АХ

MOV AX, DSEG ;помещение значения начала сегмента DSEG в регистр АХ

MOV DS, AX ;помещение в регистр DS значение регистра АХ

CALL NUMBERINPUT ;вызов процедуры NUMBERINPUT

CALL NUMBEROUTPUT ;вызов процедуры NUMBEROUTPUT

MOV AH, 8H ;помещение в регистр АН значение 8 в hex

INT 21H ;вызов 21 прерывания с функцией 8 - ввод с клавиатуры без отображения на экране

RET ;возврат из процедуры

MAIN ENDP ;конец процедуры

NUMBERINPUT PROC NEAR ;начало процедуры NUMBERINPUT, NEAR - определяет дальность возврата

MOV AH, 9H ;запись в АН значения 9

LEA DX, START\_STR ;запись в регистр адреса смещения START\_STR

INT 21H ;вызов 21 прерывания с функцией 9 - вывод строки на экран

LEA DX, DUMP ;запись в регистр DX адреса смещения DUMP

MOV AH, 0AH ;запись 10 в АН

INT 21H ;вызов 21 прерывания с функцией 0AH - ввод с клавиатуры

LEA DI, DUMP+2 ;запись в DI адрес DUMP+2

MOV AL, [DI] ;запись в AL адрес [DI]

CMP AL, '-' ;сравнение значение AL с '-'

JE MINUS ;если да - переход к части MINUS

CMP AL, '+' ;сравнение значение AL с '+'

JE PLUS ;если да - переход к части PLUS

REGISTERCLEAR: ;начало части REGISTERCLEAR

XOR BX, BX ;сравнение BX с BX

MOV BX, 10 ;запись в BX 10

XOR AX, AX ;сравнение BX с BX

ERRORCHECK: ;начало части ERRORCHECK

XOR CX, CX ;сравнение СX с СX

CMP BYTE PTR [DI], 0DH ;сравнение значения в DI с 0DH - 'Enter'

JZ STRINGEND ;перейти к STRINGEND, если значение = 0 (DI = 0DH)

CMP BYTE PTR [DI], 30H ;сравнение значения в DI с 30H - '0'

JB ERROR ;перейти к ERROR, если DI < 30H

CMP BYTE PTR [DI], 39H ;сравнение значения в DI с 39H - '9'

JA ERROR ;перейти к ERROR, если DI > 39H

MOV CL, [DI] ;переместить в CL DI

SUB CL, 30H ;вычетание из CL 30h

MUL BX ;произведение байта из AX на BX

JC ERROR ;перейти в ERROR, если был осуществлен перенос

ADD AX, CX ;сложение АХ и СХ

JC ERROR ;перейти к ERROR, если есть перенос

INC DI ;увеличивает значение DI на 1

JMP ERRORCHECK ;переход к ERRORCHECK

STRINGEND: ;начало части STRINGEND

MOV CL, NEGATIVE ;перемещение в CL значение NEGATIVE (0)

CMP CL, 1 ;сравнение CL с 1

JNZ INPUTEND ;перейти к INPUTEND, если не 0 (CL != 1)

NEG AX ;изменение знака в АХ

JMP INPUTEND ;переход к INPUTEND

PLUS: ;начало части PLUS

INC DI ;увеличение значения на DI на 1

JMP REGISTERCLEAR ;переход к REGISTERCLEAR

MINUS: ;начало части MINUS

MOV AL, 1 ;перемещение в AL 1

MOV NEGATIVE, AL ;перемещение в NEGATIVE значение AL

INC DI ;увеличение DI на 1

JMP REGISTERCLEAR ;переход к REGISTERCLEAR

ERROR: ;начало части ERROR

MOV AH, 9 ;помещение 9 в АН

LEA DX, ERROR\_STR ;запись в регистр DX адрес смещения ERROR\_STR

INT 21H ;вызов 21 прерывания с функцией 9 - вывод на экран строки

MOV AH, 8H ;перемещение 8h в АН

INT 21H ;вызов 21 прерывания с функцией 8 - ввод с клавиатуры без отображения на экране

MOV AH, 4CH ;перемещение 4h в АН

INT 21H ;вызов 21 прерывания с функцией 4с - завершение выполнения программы

INPUTEND: ;начало части INPUTEND

MOV DIGIT, AX ;перемещение в DIGIT значения АХ

RET ;возврат из процедуры

NUMBERINPUT ENDP ;конец процедуры NUMBERINPUT

NUMBEROUTPUT PROC NEAR ;начало процедуры NUMBEROUTPUT, NEAR - определяет дальность возврата

XOR BX, BX ;сравнение BX с BX

XOR AX, AX ;сравнение AX с AX

MOV AH, 9 ;перемещение 9 в регистр АН

LEA DX, OUTPUT\_STR ;запись в регистр DX адрес смещения OUTPUT

INT 21H ;вызов 21 прерывания с функцией 9 - вывод на экран строки

MOV BX, DIGIT ;перемещение значение DIGIT в BX

OR BX, BX ;сравнение BX c BX

JNS POSITIVECASE ;перейти к POSITIVECASE, если не установлен флаг знака

MOV AH, 2 ;перемещение 2 в регистр АН

MOV AL, '-' ;перемещ5ение '-' в регистр AL

INT 29H ;вызов 29 прерывания с функцией 2 - вывод символа на экран

NEG BX ;изменение знака в ВХ

POSITIVECASE: ;начало части POSITIVECASE

MOV AX, BX ;перемещение в АХ значение ВХ

XOR CX, CX ;сравнение Сх с СХ

MOV BX, 10 ;перемещение 10 в регистр ВХ

LOOP1: ;начало части LOOP1

XOR DX, DX ;сравнение DX c DX

DIV BX ;делит AL на BX

ADD DL, 30H ;добавляет к DL 30H

PUSH DX ;помещение DX в стек

INC CX ;увеличивает значение CX на 1

TEST AX, AX ;проверка AX c AX

JNZ LOOP1 ;переход к LOOP1, если равны

LOOP2: ;начало части LOOP2

POP AX ;извлечь АХ

INT 29H ;вызов 29 прерывания

LOOP LOOP2 ;повторять LOOP2 до завершения

RET ;возврат из процедуры

NUMBEROUTPUT ENDP ;конец процедуры

CSEG ENDS ;конец сегмента кода

END MAIN ;конец программы

**Введені та отримані результати:**

**Вміст .lst файлу:**

Turbo Assembler Version 2.02 10/18/17 23:35:36 Page 1

lab2.ASM

1 0000 STSEG SEGMENT PARA STACK "STACK" ;объявление сегмента стека. stseg - имя +

2 сегмента, segment - зарезервированное слово, para - тип выравнивания, кратное 16, stack - +

3 необходимо для обэъединения, "stack" - слово, по которому будет выравниваться

4 0000 40\*(53 54 41 43 4B) DB 64 DUP ("STACK") ;определение 64 слов "stack"

5 0140 STSEG ENDS ;конец объявления сегмента стека

6

7 0000 dseg segment para public 'data' ;объявление сегмента данных.

8 0000 45 4E 54 45 52 20 4E+ START\_STR DB "ENTER NUMBER (-999 TO 9999): ", 13, 10, '$' ;объявление строки, 13 - конец +

9 55 4D 42 45 52 20 28+ строки, 10 -

10 2D 39 39 39 20 54 4F+

11 20 39 39 39 39 29 3A+

12 20 0D 0A 24

13 0020 49 4E 43 4F 52 52 45+ ERROR\_STR DB "INCORRECT INPUT", 13, 10, '$' ;объявление строки

14 43 54 20 49 4E 50 55+

15 54 0D 0A 24

16 0032 45 4E 54 45 52 45 44+ OUTPUT\_STR DB "ENTERED NUMBER: ", 13, 10, '$' ;объявление строки

17 20 4E 55 4D 42 45 52+

18 3A 20 0D 0A 24

19 0045 05 ?? 05\*(24) DUMP DB 5, ?, 5 DUP("$") ;запись в dump 5, ?, и 5 раз ?

20 004C 0000 DIGIT DW 0 ;объявление значения равно 0, +

21 DW - 2 байта

22 004E 00 NEGATIVE DB 0 ;объявление знака равно 0, DB -+

23 1 байт

24 004F DSEG ENDS ;конец объявление сегмета данных

25

26 0000 CSEG SEGMENT PARA PUBLIC "CODE" ;начало объявления сегмента кода

27

28 0000 MAIN PROC FAR ;начало выполнения процедуры +

29 "main", far - определяет дальность возврата

30 ASSUME CS:CSEG,DS:DSEG,SS:STSEG ;привязка сегментов к регистрам

31

32 0000 1E PUSH DS ;помещение DS в стек

33 0001 50 PUSH AX ;помещение AX в стек

34 0002 B8 0000 MOV AX, 0 ;помещение 0 в регистр АХ

35 0005 B8 0000s MOV AX, DSEG ;помещение значения начала +

36 сегмента DSEG в регистр АХ

37 0008 8E D8 MOV DS, AX ;помещение в регистр DS +

38 значение регистра АХ

39

40 000A E8 0008 CALL NUMBERINPUT ;вызов процедуры NUMBERINPUT

41 000D E8 0074 CALL NUMBEROUTPUT ;вызов процедуры NUMBEROUTPUT

42

43 0010 B4 08 MOV AH, 8H ;помещение в регистр АН +

44 значение 8 в hex

45 0012 CD 21 INT 21H ;вызов 21 прерывания с функцией+

46 8 - ввод с клавиатуры без отображения на экране

47

48 0014 CB RET ;возврат из процедуры

49 0015 MAIN ENDP ;конец процедуры

50

51

52 0015 NUMBERINPUT PROC NEAR ;начало процедуры NUMBERINPUT, +

53 NEAR - определяет дальность возврата

54

55 0015 B4 09 MOV AH, 9H ;запись в АН значения 9

56 0017 BA 0000r LEA DX, START\_STR ;запись в регистр адреса +

57 смещения START\_STR

Turbo Assembler Version 2.02 10/18/17 23:35:36 Page 2

lab2.ASM

58 001A CD 21 INT 21H ;вызов 21 прерывания с функцией+

59 9 - вывод строки на экран

60

61 001C BA 0045r LEA DX, DUMP ;запись в регистр DX адреса +

62 смещения DUMP

63 001F B4 0A MOV AH, 0AH ;запись 10 в АН

64 0021 CD 21 INT 21H ;вызов 21 прерывания с функцией+

65 0AH - ввод с клавиатуры

66

67 0023 BF 0047r LEA DI, DUMP+2 ;запись в DI адрес DUMP+2

68 0026 8A 05 MOV AL, [DI] ;запись в AL адрес [DI]

69 0028 3C 2D CMP AL, '-' ;сравнение значение AL с '-'

70 002A 74 3D JE MINUS ;если да - переход к части MINUS

71 002C 3C 2B CMP AL, '+' ;сравнение значение AL с '+'

72 002E 74 36 JE PLUS ;если да - переход к части PLUS

73

74 0030 REGISTERCLEAR: ;начало части REGISTERCLEAR

75 0030 33 DB XOR BX, BX ;сравнение BX с BX

76 0032 BB 000A MOV BX, 10 ;запись в BX 10

77 0035 33 C0 XOR AX, AX ;сравнение BX с BX

78

79 0037 ERRORCHECK: ;начало части ERRORCHECK

80 0037 33 C9 XOR CX, CX ;сравнение СX с СX

81 0039 80 3D 0D CMP BYTE PTR [DI], 0DH ;сравнение значения в DI с 0DH +

82 - 'Enter'

83 003C 74 1A JZ STRINGEND ;перейти к STRINGEND, если +

84 значение = 0 (DI = 0DH)

85 003E 80 3D 30 CMP BYTE PTR [DI], 30H ;сравнение значения в DI с 30H +

86 - '0'

87 0041 72 2E JB ERROR ;перейти к ERROR, если DI < 30H

88 0043 80 3D 39 CMP BYTE PTR [DI], 39H ;сравнение значения в DI с 39H +

89 - '9'

90 0046 77 29 JA ERROR ;перейти к ERROR, если DI > 39H

91 0048 8A 0D MOV CL, [DI] ;переместить в CL DI

92 004A 80 E9 30 SUB CL, 30H ;вычетание из CL 30h

93 004D F7 E3 MUL BX ;произведение байта из AX на BX

94 004F 72 20 JC ERROR ;перейти в ERROR, если был +

95 осуществлен перенос

96 0051 03 C1 ADD AX, CX ;сложение АХ и СХ

97 0053 72 1C JC ERROR ;перейти к ERROR, если есть +

98 перенос

99 0055 47 INC DI ;увеличивает значение DI на 1

100 0056 EB DF JMP ERRORCHECK ;переход к ERRORCHECK

101

102 0058 STRINGEND: ;начало части STRINGEND

103 0058 8A 0E 004Er MOV CL, NEGATIVE ;перемещение в CL значение +

104 NEGATIVE (0)

105 005C 80 F9 01 CMP CL, 1 ;сравнение CL с 1

106 005F 75 1F JNZ INPUTEND ;перейти к INPUTEND, если не 0 +

107 (CL != 1)

108 0061 F7 D8 NEG AX ;изменение знака в АХ

109 0063 EB 1B 90 JMP INPUTEND ;переход к INPUTEND

110

111 0066 PLUS: ;начало части PLUS

112 0066 47 INC DI ;увеличение значения на DI на 1

113 0067 EB C7 JMP REGISTERCLEAR ;переход к REGISTERCLEAR

114

Turbo Assembler Version 2.02 10/18/17 23:35:36 Page 3

lab2.ASM

115 0069 MINUS: ;начало части MINUS

116 0069 B0 01 MOV AL, 1 ;перемещение в AL 1

117 006B A2 004Er MOV NEGATIVE, AL ;перемещение в NEGATIVE +

118 значение AL

119 006E 47 INC DI ;увеличение DI на 1

120 006F EB BF JMP REGISTERCLEAR ;переход к REGISTERCLEAR

121

122 0071 ERROR: ;начало части ERROR

123 0071 B4 09 MOV AH, 9 ;помещение 9 в АН

124 0073 BA 0020r LEA DX, ERROR\_STR ;запись в регистр DX адрес +

125 смещения ERROR\_STR

126 0076 CD 21 INT 21H ;вызов 21 прерывания с функцией+

127 9 - вывод на экран строки

128

129 0078 B4 08 MOV AH, 8H ;перемещение 8h в АН

130 007A CD 21 INT 21H ;вызов 21 прерывания с функцией+

131 8 - ввод с клавиатуры без отображения на экране

132

133 007C B4 4C MOV AH, 4CH ;перемещение 4h в АН

134 007E CD 21 INT 21H ;вызов 21 прерывания с функцией+

135 4с - завершение выполнения программы

136

137 0080 INPUTEND: ;начало части INPUTEND

138 0080 A3 004Cr MOV DIGIT, AX ;перемещение в DIGIT значения АХ

139 0083 C3 RET ;возврат из процедуры

140

141 0084 NUMBERINPUT ENDP ;конец процедуры NUMBERINPUT

142

143 0084 NUMBEROUTPUT PROC NEAR ;начало процедуры NUMBEROUTPUT,+

144 NEAR - определяет дальность возврата

145

146 0084 33 DB XOR BX, BX ;сравнение BX с BX

147 0086 33 C0 XOR AX, AX ;сравнение AX с AX

148

149 0088 B4 09 MOV AH, 9 ;перемещение 9 в регистр АН

150 008A BA 0032r LEA DX, OUTPUT\_STR ;запись в регистр DX адрес +

151 смещения OUTPUT

152 008D CD 21 INT 21H ;вызов 21 прерывания с функцией+

153 9 - вывод на экран строки

154

155 008F 8B 1E 004Cr MOV BX, DIGIT ;перемещение значение DIGIT в BX

156 0093 0B DB OR BX, BX ;сравнение BX c BX

157 0095 79 08 JNS POSITIVECASE ;перейти к POSITIVECASE, если +

158 не установлен флаг знака

159

160 0097 B4 02 MOV AH, 2 ;перемещение 2 в регистр АН

161 0099 B0 2D MOV AL, '-' ;перемещ5ение '-' в регистр AL

162 009B CD 29 INT 29H ;вызов 29 прерывания с функцией+

163 2 - вывод символа на экран

164

165 009D F7 DB NEG BX ;изменение знака в ВХ

166

167 009F POSITIVECASE: ;начало части POSITIVECASE

168 009F 8B C3 MOV AX, BX ;перемещение в АХ значение ВХ

169 00A1 33 C9 XOR CX, CX ;сравнение Сх с СХ

170 00A3 BB 000A MOV BX, 10 ;перемещение 10 в регистр ВХ

171

Turbo Assembler Version 2.02 10/18/17 23:35:36 Page 4

lab2.ASM

172 00A6 LOOP1: ;начало части LOOP1

173 00A6 33 D2 XOR DX, DX ;сравнение DX c DX

174 00A8 F7 F3 DIV BX ;делит AL на BX

175 00AA 80 C2 30 ADD DL, 30H ;добавляет к DL 30H

176 00AD 52 PUSH DX ;помещение DX в стек

177 00AE 41 INC CX ;увеличивает значение CX на 1

178

179 00AF 85 C0 TEST AX, AX ;проверка AX c AX

180 00B1 75 F3 JNZ LOOP1 ;переход к LOOP1, если равны

181

182 00B3 LOOP2: ;начало части LOOP2

183 00B3 58 POP AX ;извлечь АХ

184 00B4 CD 29 INT 29H ;вызов 29 прерывания

185 00B6 E2 FB LOOP LOOP2 ;повторять LOOP2 до завершения

186

187 00B8 C3 RET ;возврат из процедуры

188 00B9 NUMBEROUTPUT ENDP ;конец процедуры

189

190 00B9 CSEG ENDS ;конец сегмента кода

191 END MAIN ;конец программы

Turbo Assembler Version 2.02 10/18/17 23:35:36 Page 5

Symbol Table

Symbol Name Type Value

??DATE Text "10/18/17"

??FILENAME Text "lab2 "

??TIME Text "23:35:36"

??VERSION Number 0202

@CPU Text 0101H

@CURSEG Text CSEG

@FILENAME Text LAB2

@WORDSIZE Text 2

DIGIT Word DSEG:004C

DUMP Byte DSEG:0045

ERROR Near CSEG:0071

ERRORCHECK Near CSEG:0037

ERROR\_STR Byte DSEG:0020

INPUTEND Near CSEG:0080

LOOP1 Near CSEG:00A6

LOOP2 Near CSEG:00B3

MAIN Far CSEG:0000

MINUS Near CSEG:0069

NEGATIVE Byte DSEG:004E

NUMBERINPUT Near CSEG:0015

NUMBEROUTPUT Near CSEG:0084

OUTPUT\_STR Byte DSEG:0032

PLUS Near CSEG:0066

POSITIVECASE Near CSEG:009F

REGISTERCLEAR Near CSEG:0030

START\_STR Byte DSEG:0000

STRINGEND Near CSEG:0058

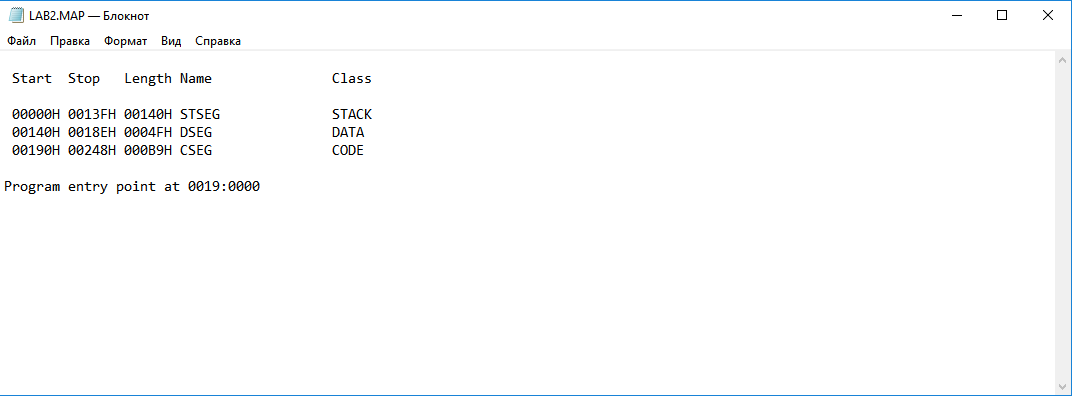
Groups & Segments Bit Size Align Combine Class

CSEG 16 00B9 Para Public CODE

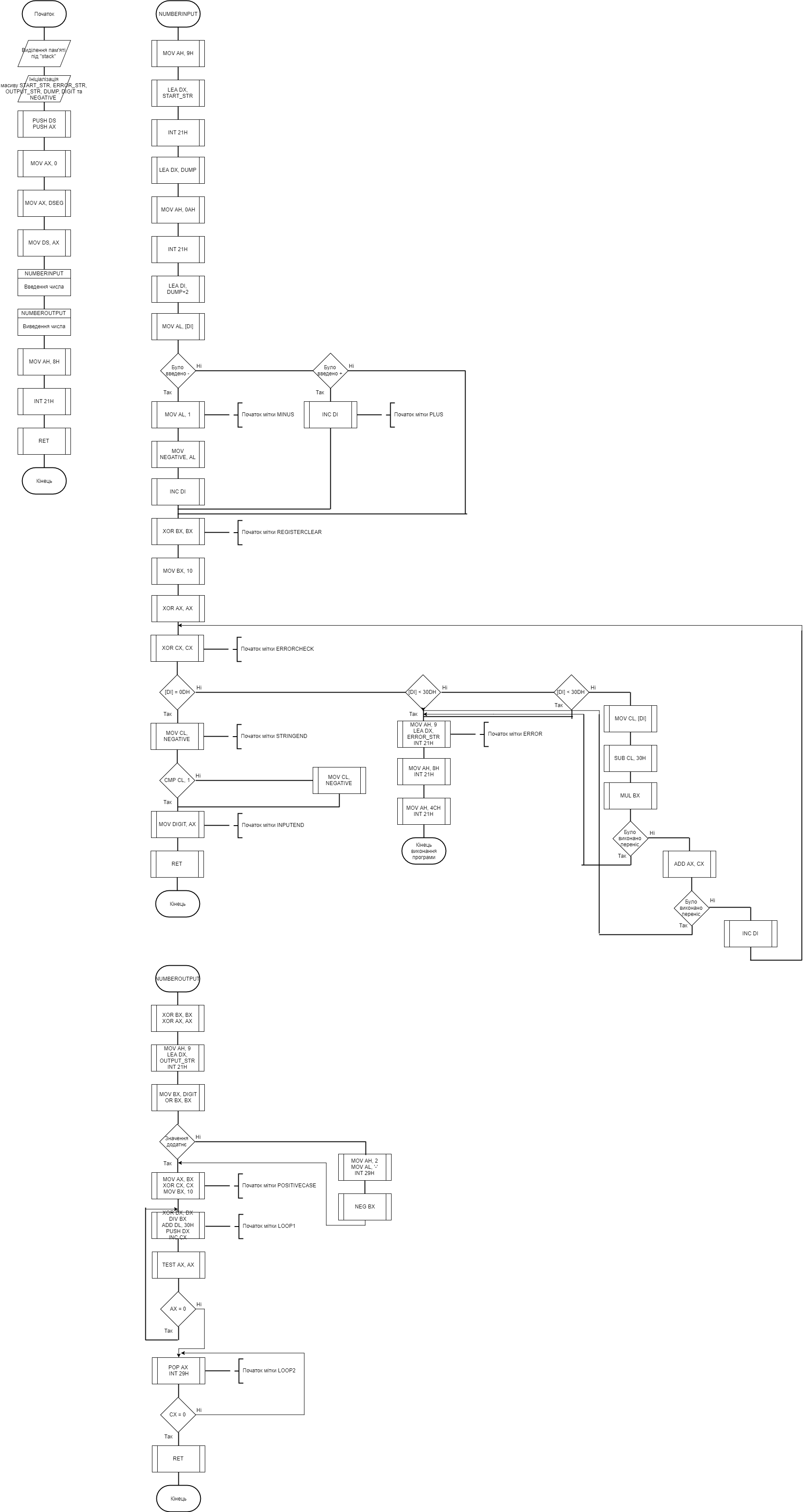
DSEG 16 004F Para Public DATA

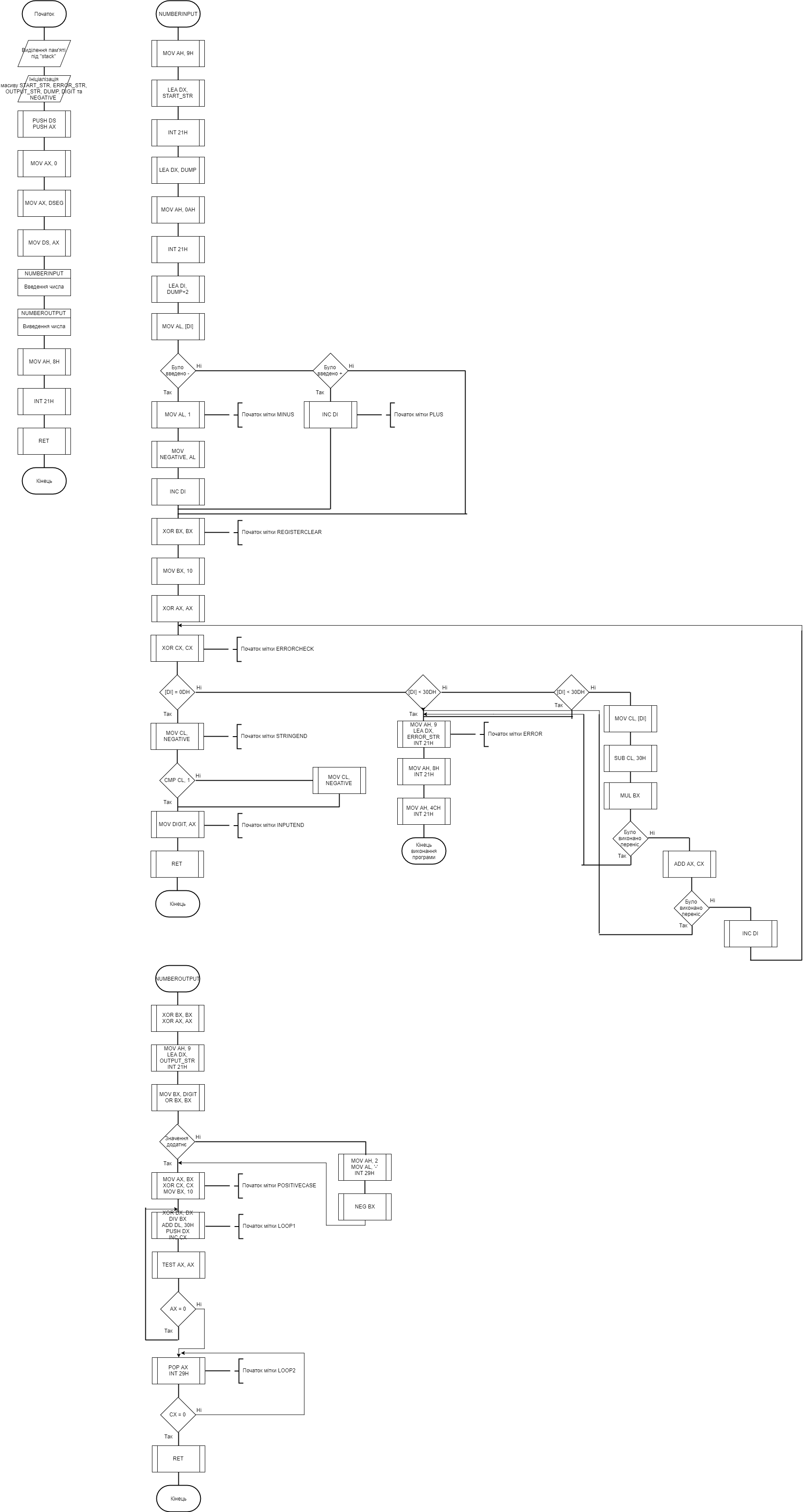
STSEG 16 0140 Para Stack STACK

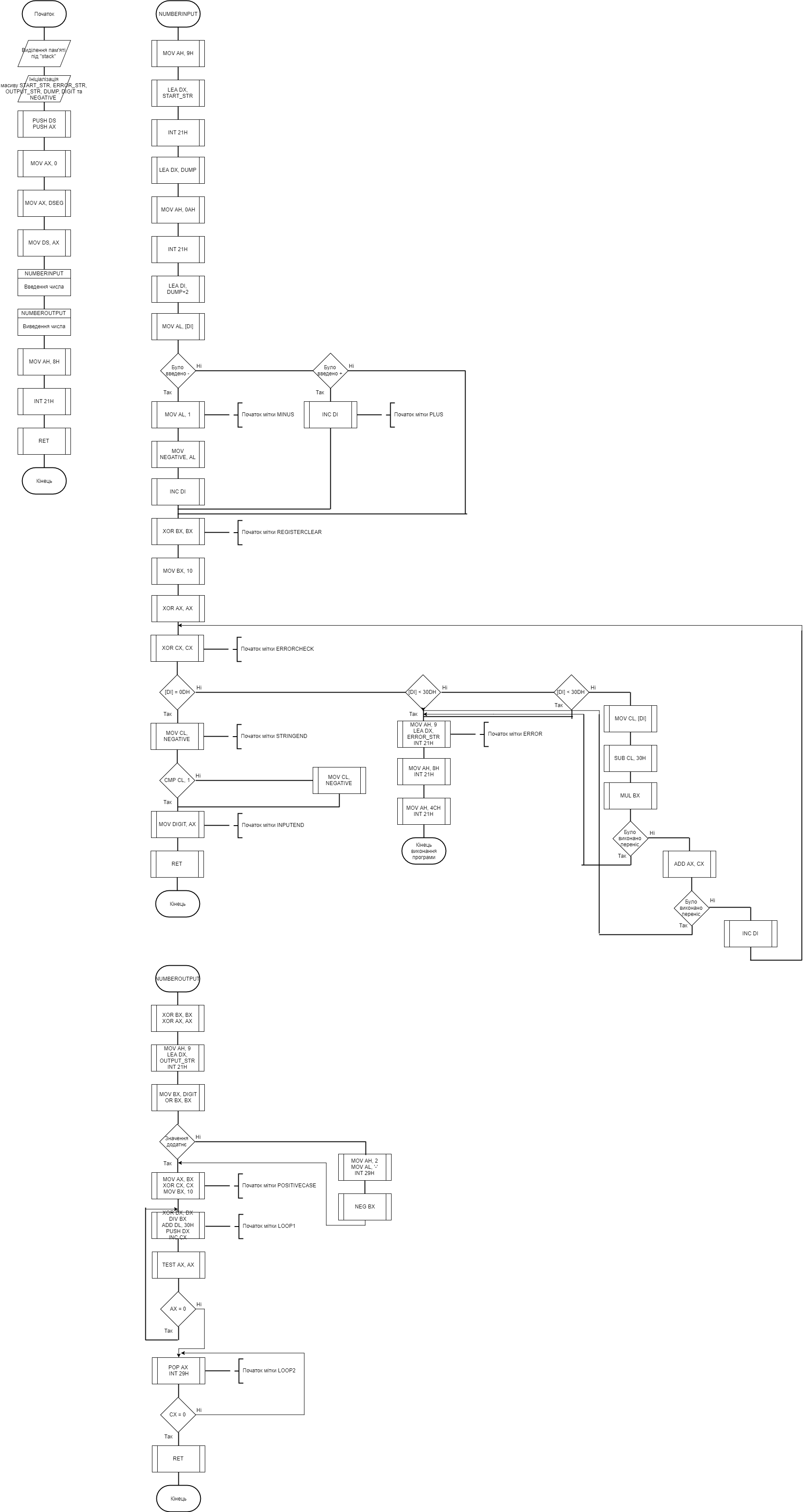
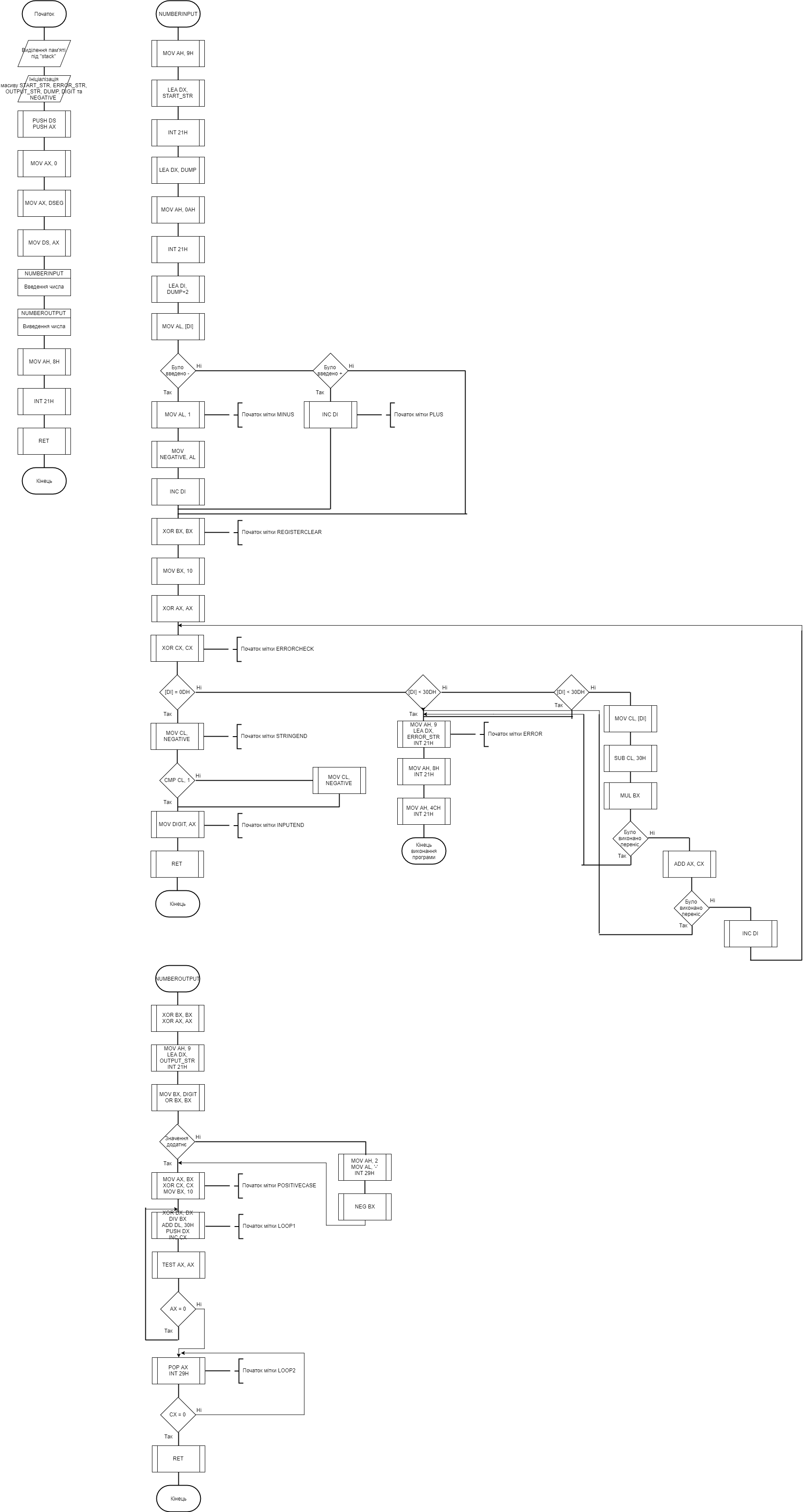
**Вміст .map файлу:**

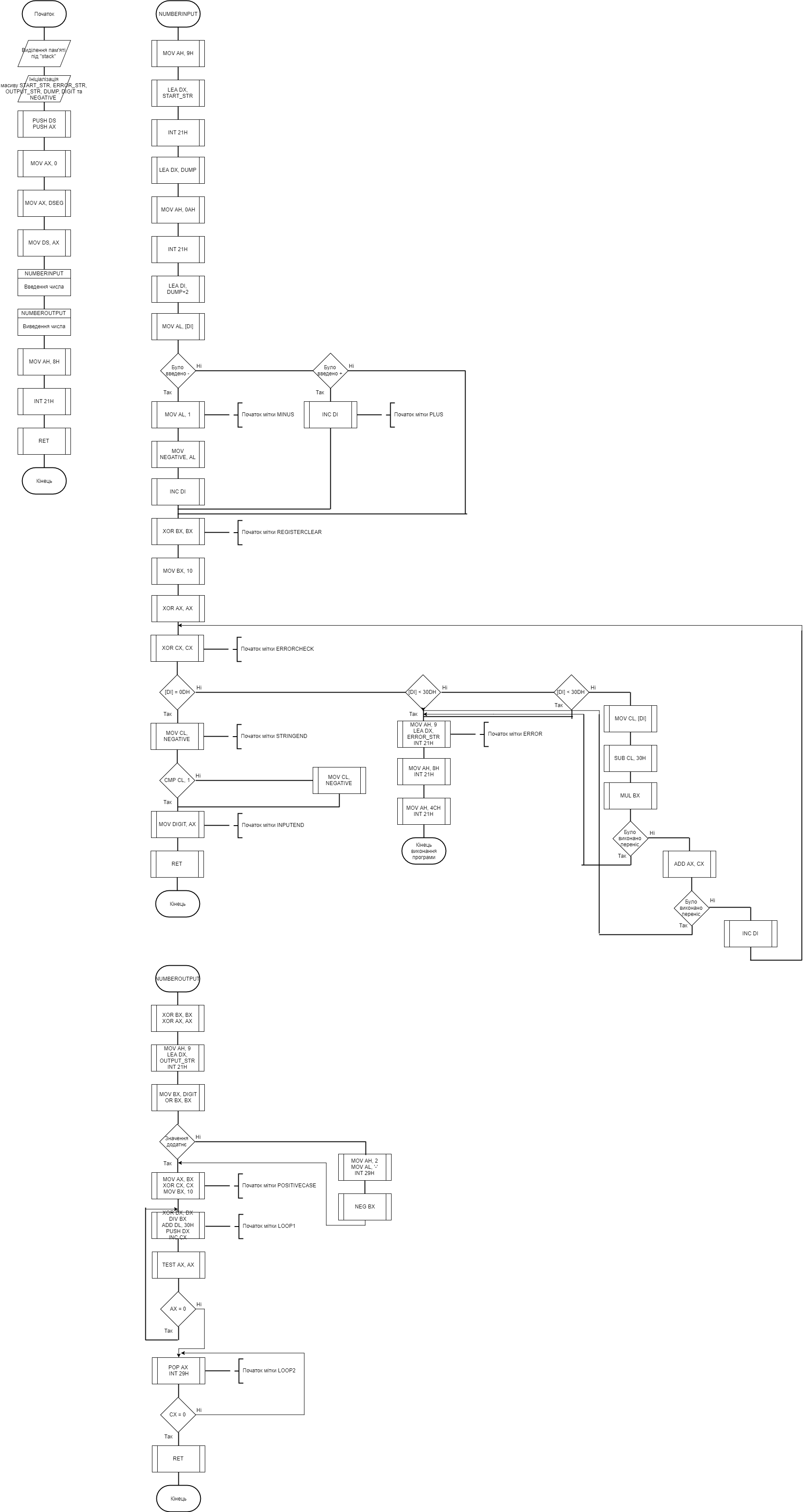
****

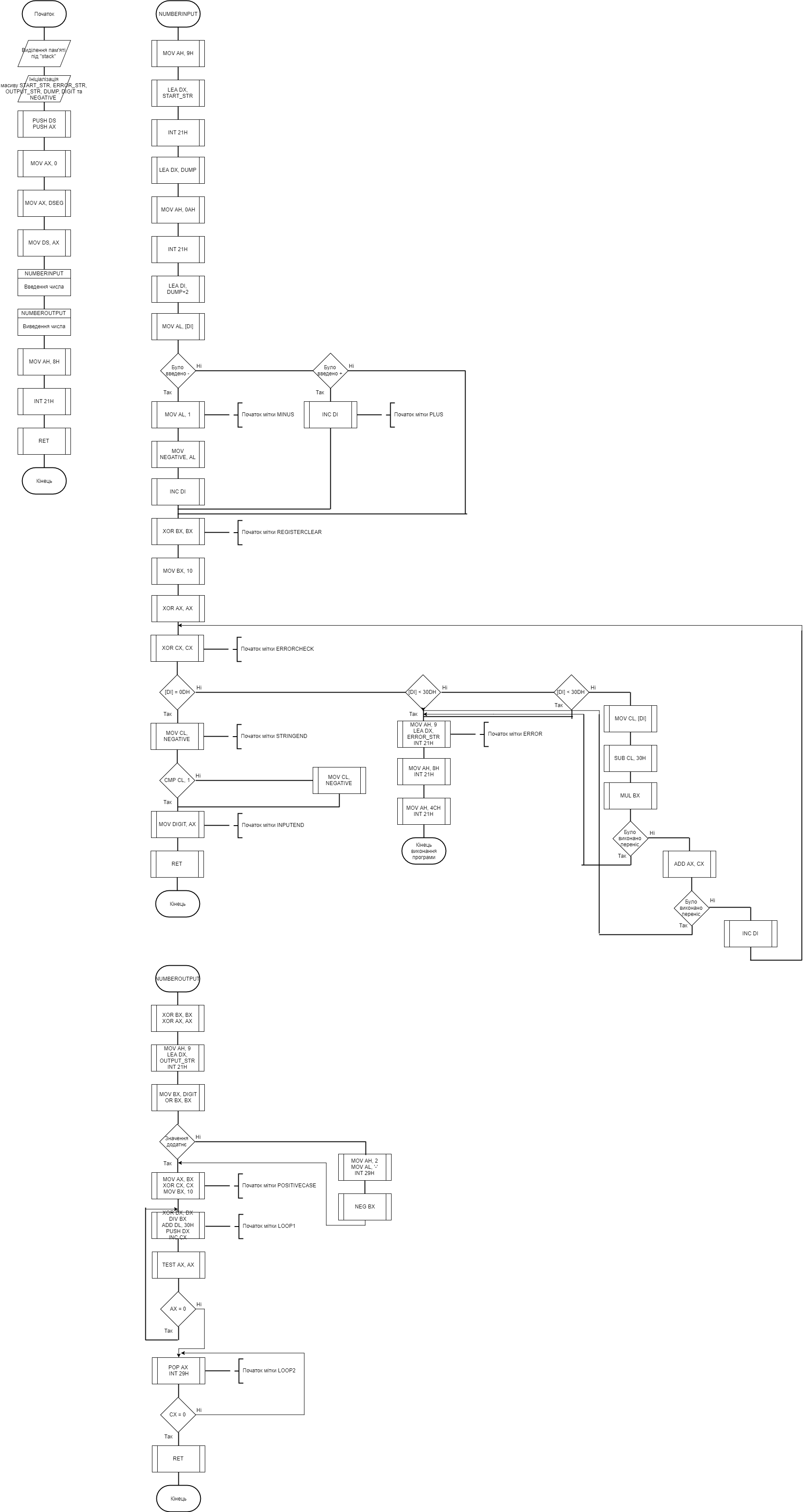
**Схема функціонування програми**





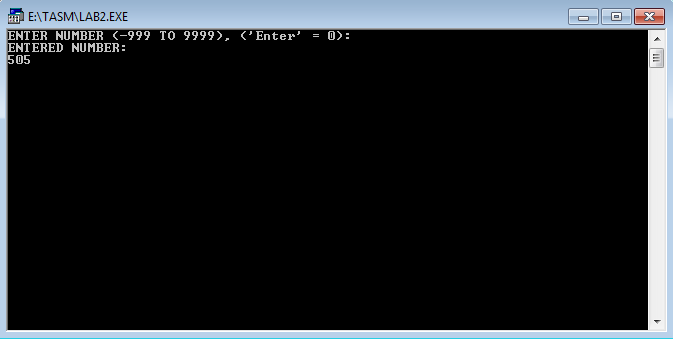
****

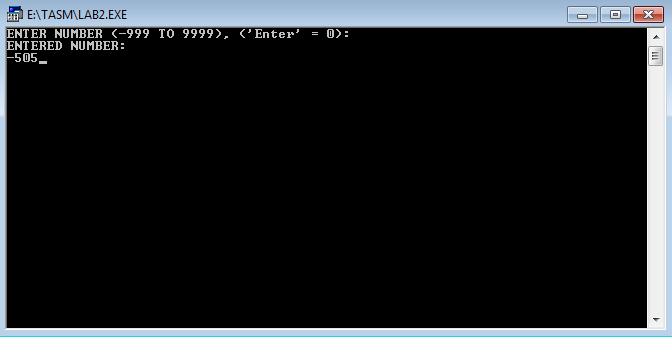
****

****

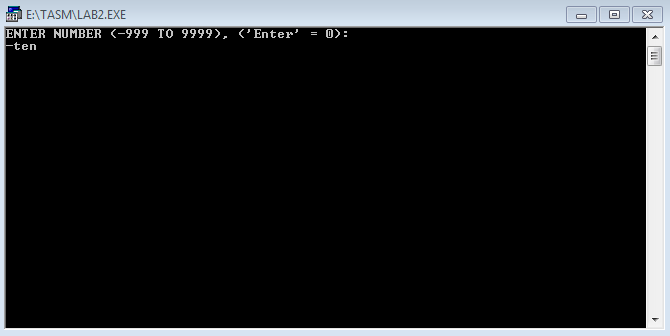
**Вікно роботи програми**

Після вводу «505»

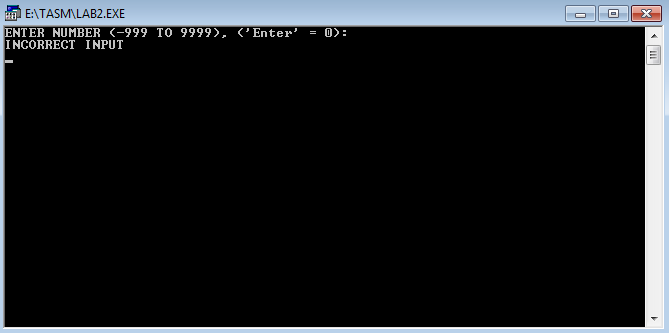


Після вводу «-505»****

1. Ввід «-ten»



1. Вивід помилки



**Висновок:**

1. Було успішно створено та протестовано процедуру виведення символу(-ів).
2. Було успішно створено та протестовано процедуру вводу рядку символів.
3. Було успішно створено та протестовано процедуру виведення цілого числа з або без знаку.
4. Було успішно створено та протестовано процедуру перетворення символу у цифру.
5. Було успішно створено та протестовано процедуру перетворення цифру у символ.
6. Було успішно створено та протестовано процедуру перевірки вводу значення.