**Комп’ютерний практикум №5**

**Тема**: макрозасоби мови асемблер.

**Завдання**:

1) переписати Програму 2.1 з використанням макросів;

2) переписати Програму 3.1, з використанням макросів;

3) переписати Програму 4.1 з використанням макросів.

**Завдання 1**:

**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), ('Enter' = 0): ", 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

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

LOOPM MACRO

LOCAL LOOP1, LOOP2

LOOP1: ;начало метки 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 метки LOOP2

POP AX ;извлечь АХ

INT 29H ;вызов 29 прерывания - вывод символа

LOOP LOOP2

ENDM

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

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

LOCAL POSITIVECASE

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, '-' ;перемещение '-' в регистр AL

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

NEG BX ;изменение знака в ВХ

POSITIVECASE: ;начало метки POSITIVECASE

MOV AX, BX ;перемещение в АХ значение ВХ

XOR CX, CX ;сравнение СX с СХ

MOV BX, 10 ;перемещение 10 в регистр ВХ

LOOPM ` ;повторять LOOP2 до завершения (CX = 0)

ENDM ;возврат из процедуры

;конец процедуры

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

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

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 0AH 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 значение по адресу DI

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

JE MINUS ;если да - переход к метки MINUS

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

JE PLUS ;если да - переход к метки PLUS

REGISTERCLEAR: ;начало метки REGISTERCLEAR

XOR BX, BX ;сравнение BX с BX

MOV BX, 10 ;запись в BX 10

XOR AX, AX ;сравнение BX с BX

ERRORCHECK: ;начало метки ERRORCHECK

XOR CX, CX ;сравнение СX с СX

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

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

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

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

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

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

MOV CL, [DI] ;переместить в CL значение по адресу 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 метки 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 метки PLUS

INC DI ;увеличение значения на DI на 1

JMP REGISTERCLEAR ;переход к REGISTERCLEAR

MINUS: ;начало метки 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 метки 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 метки INPUTEND

MOV DIGIT, AX ;перемещение в DIGIT значения АХ

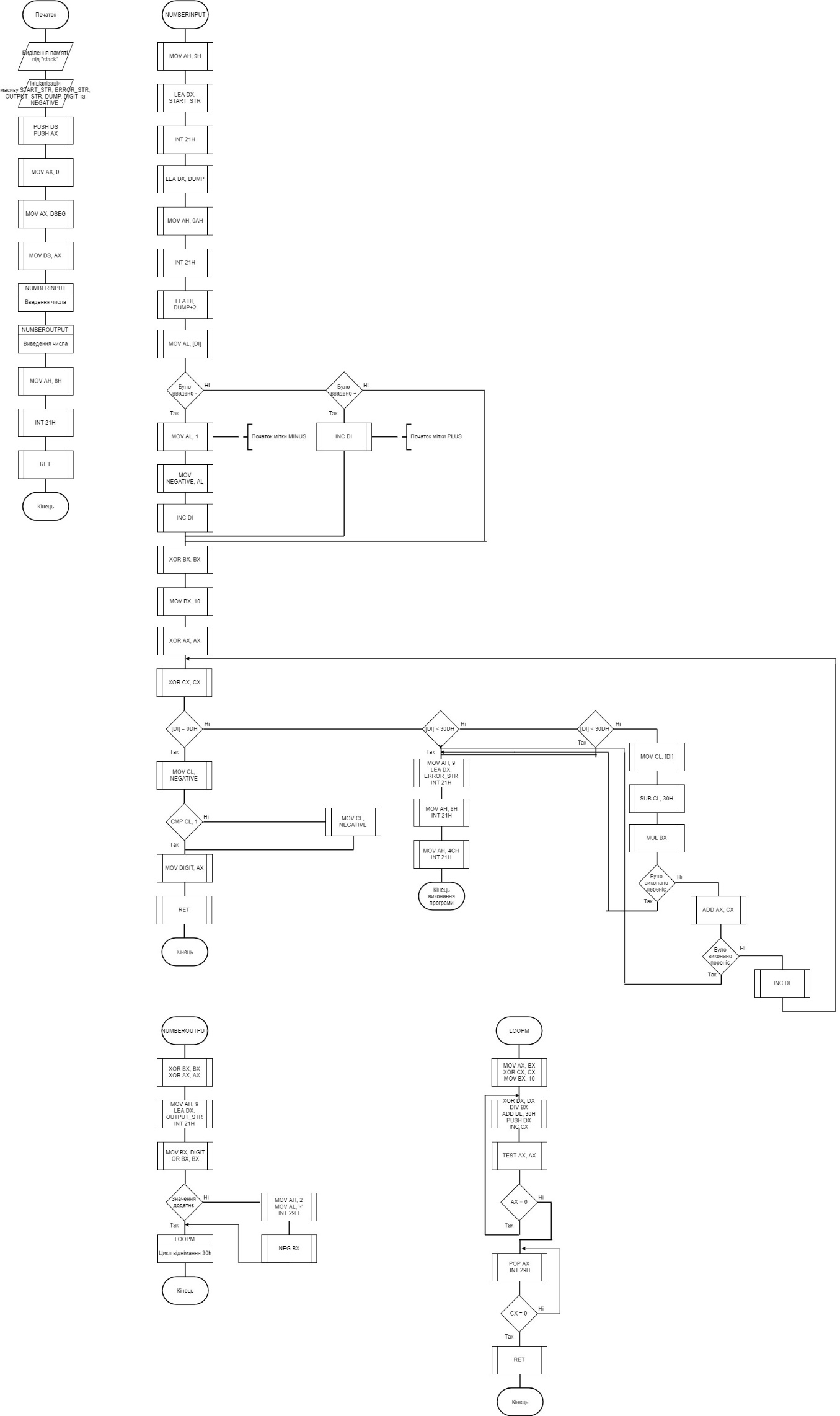
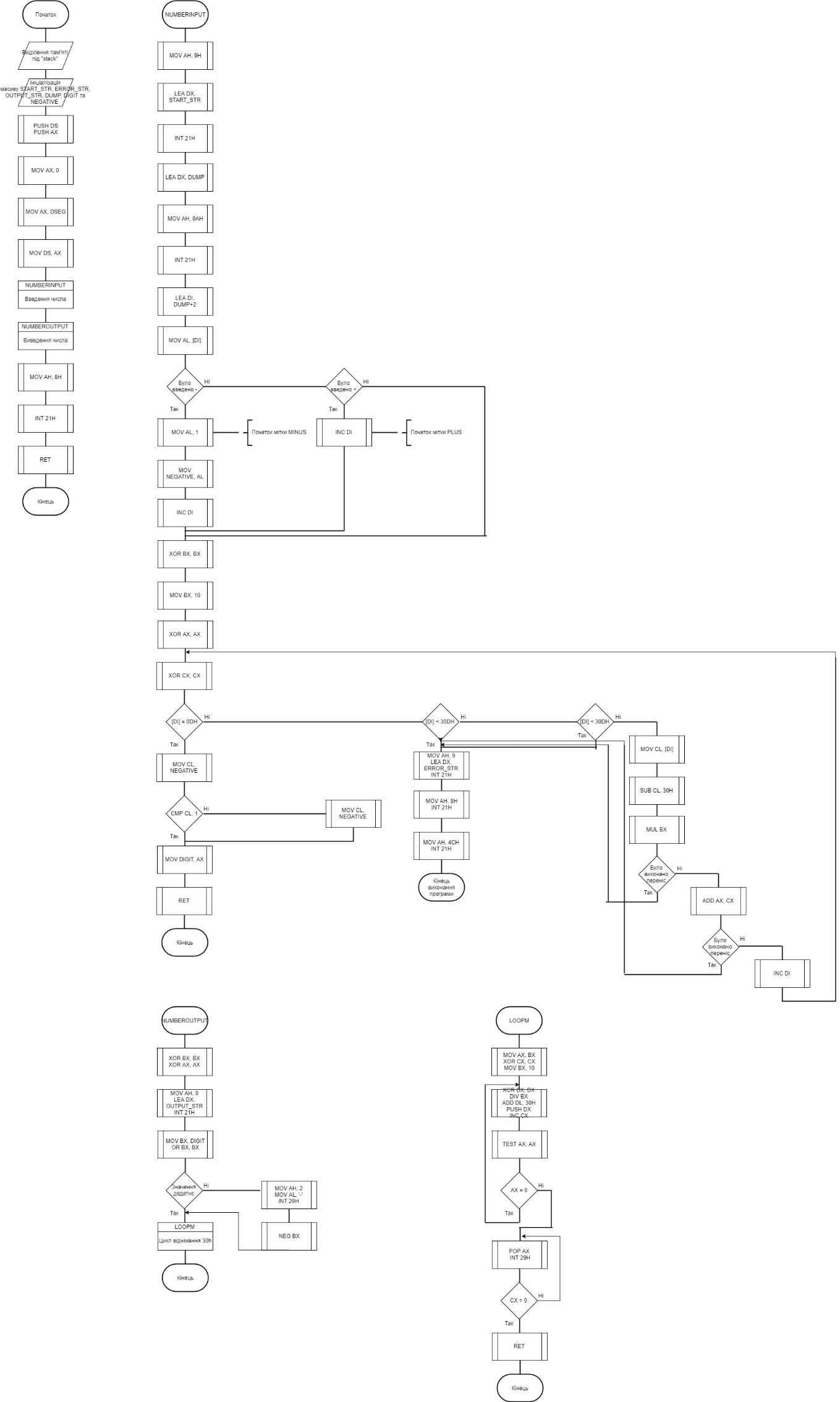
RET ;возврат из процедуры

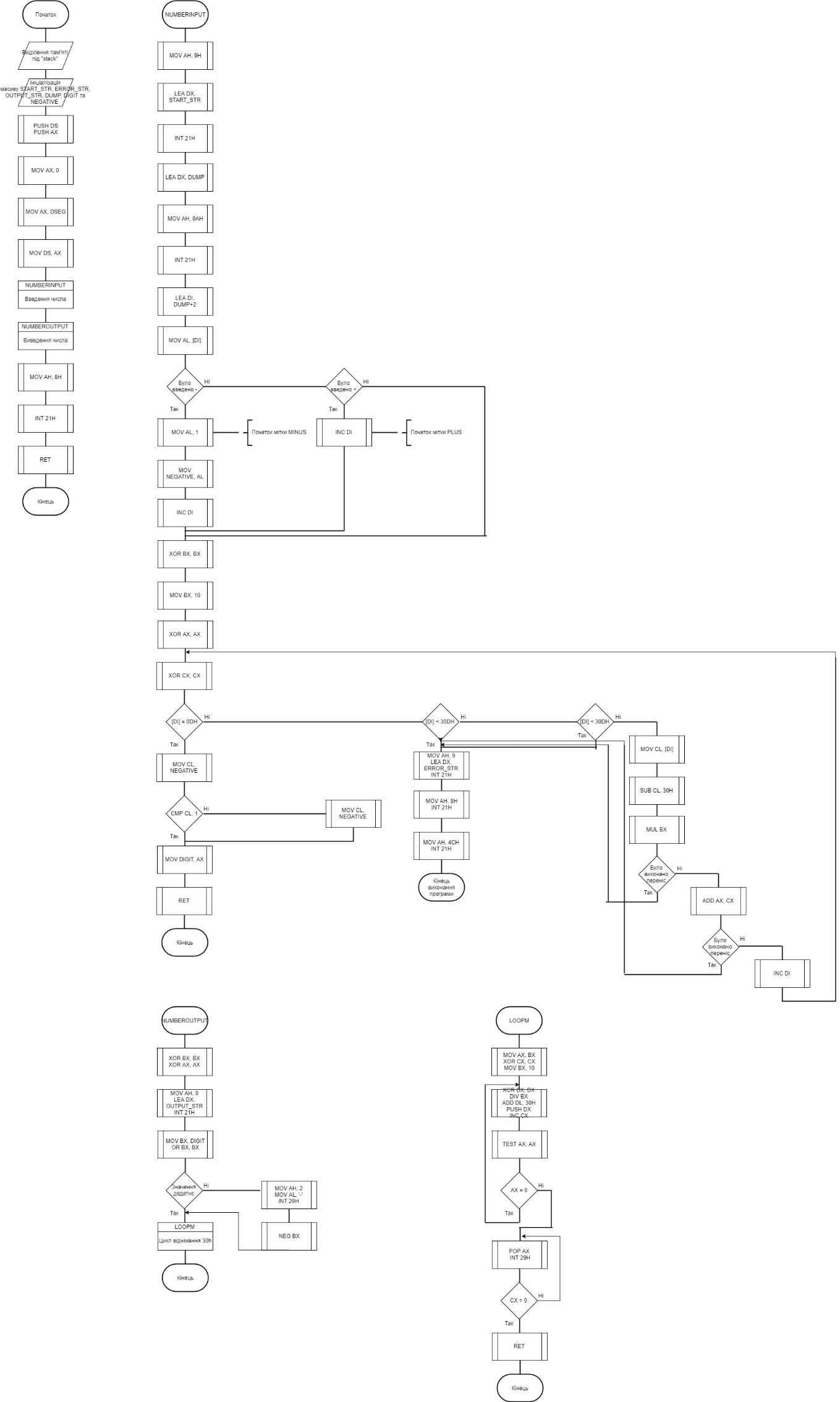
NUMBERINPUT ENDP ;конец процедуры NUMBERINPUT

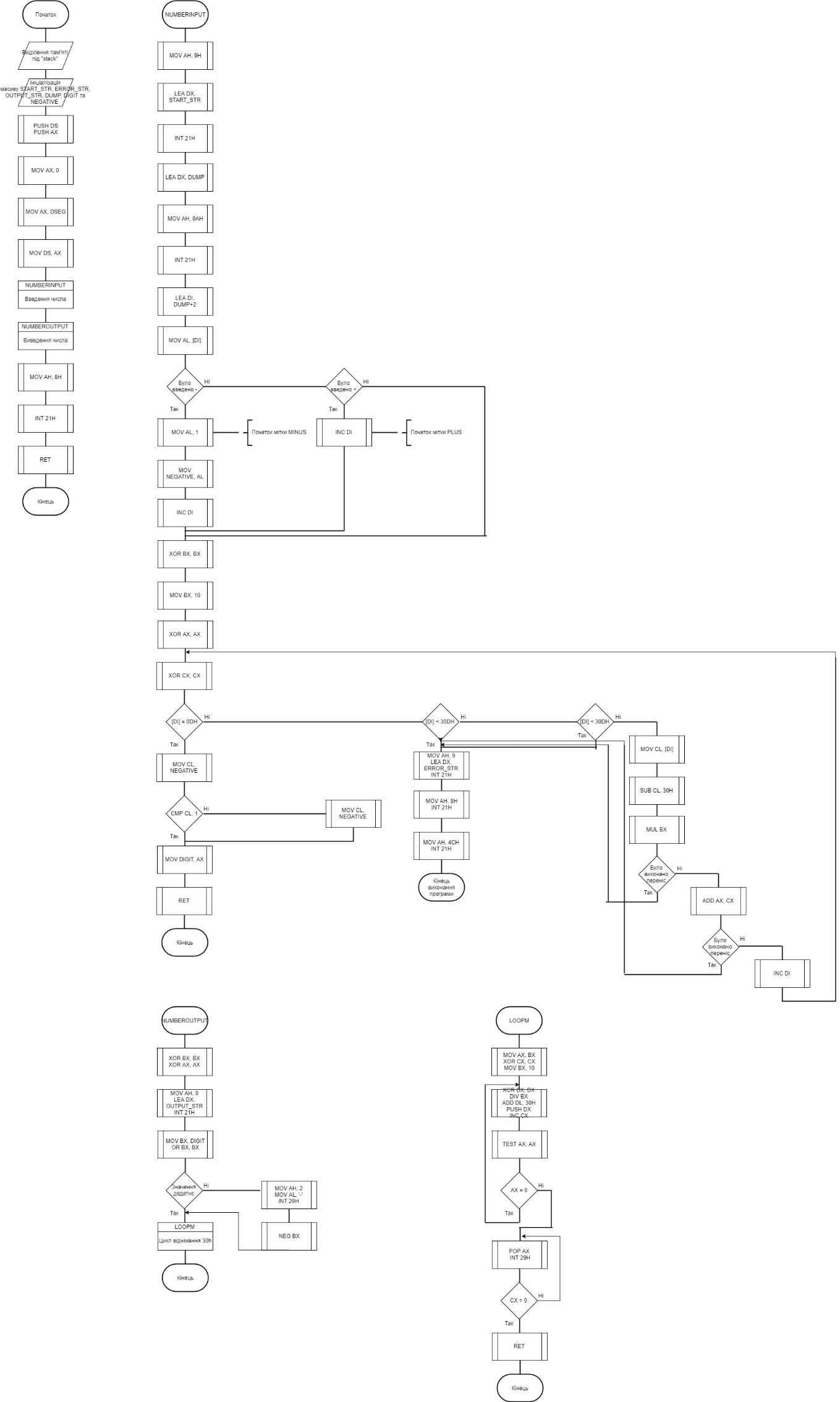
CSEG ENDS ;конец сегмента кода

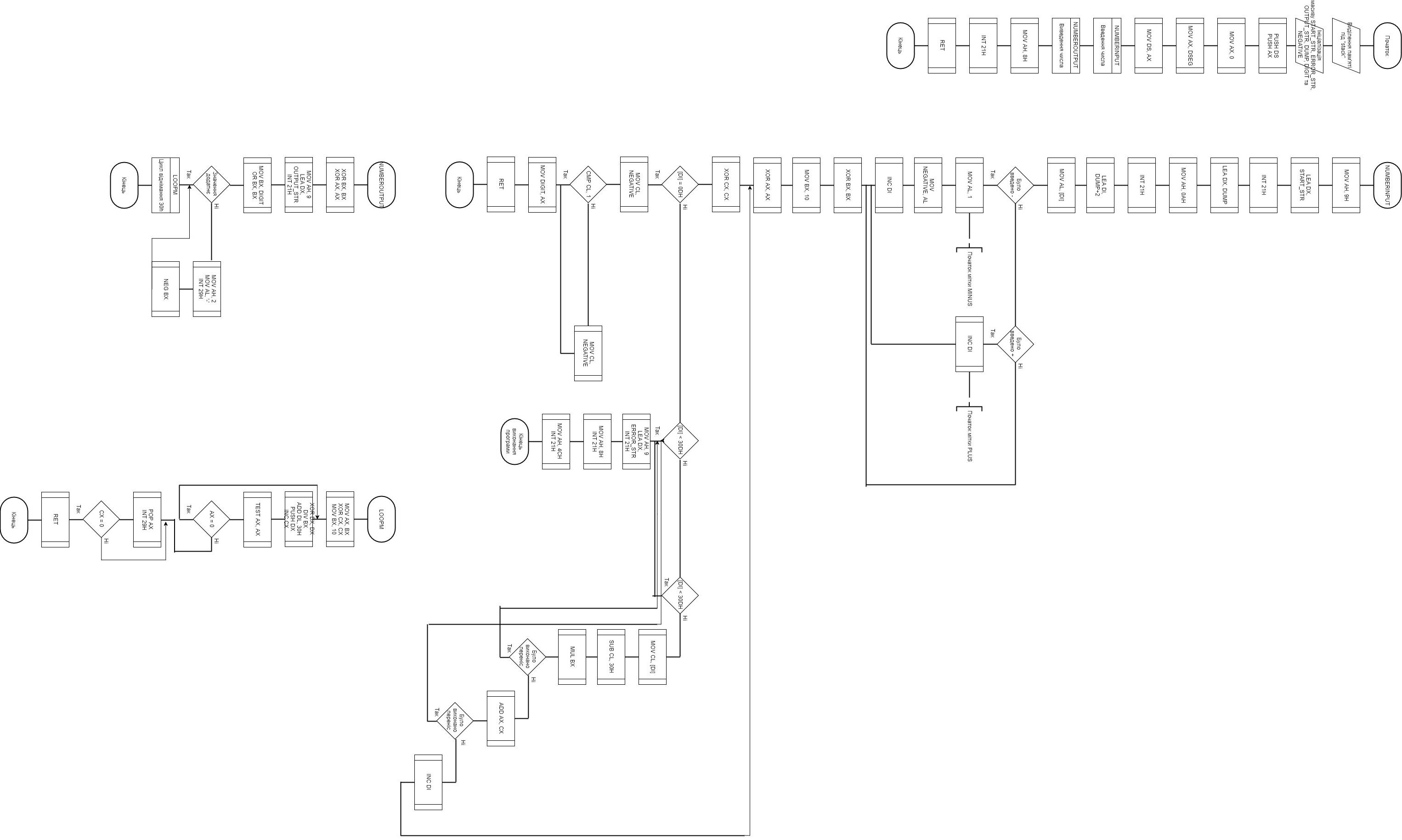
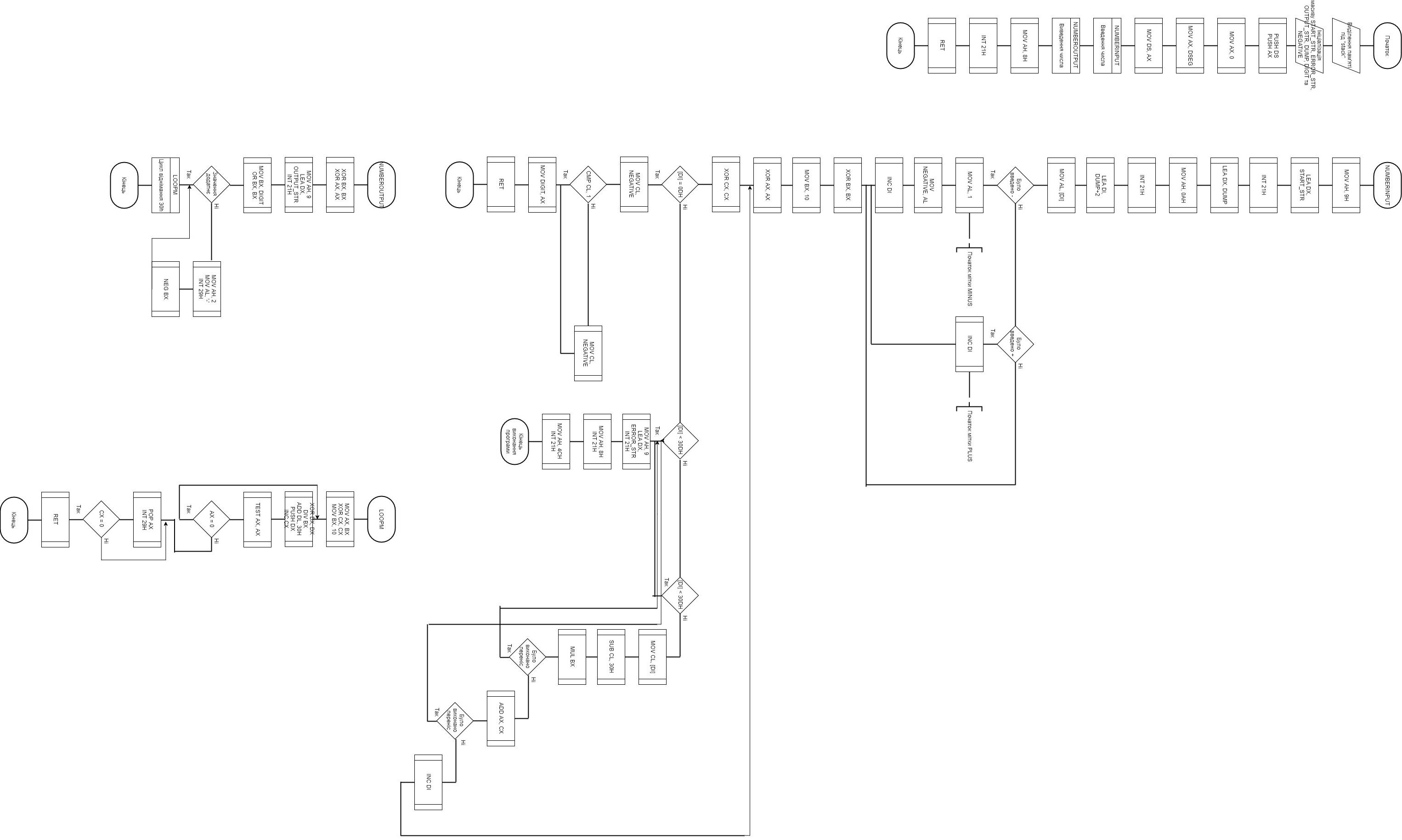
END MAIN ;конец программы

**Схема функціонування програми**

****

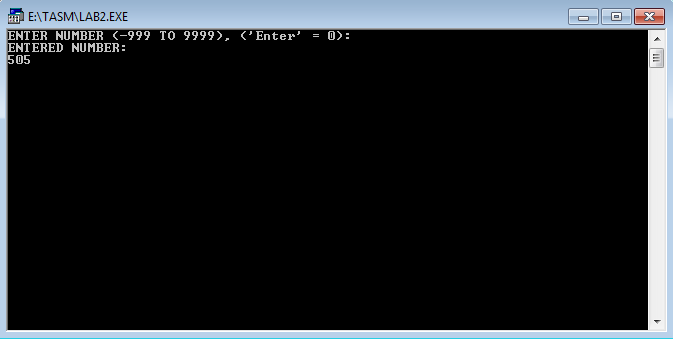
****

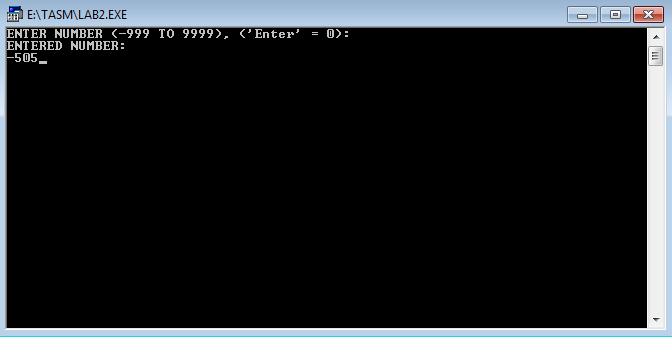
****



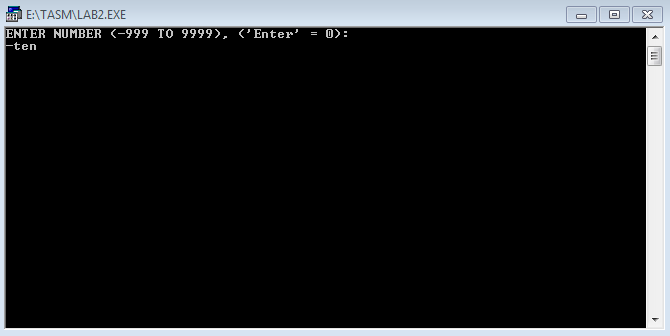
**Вікно роботи програми**

Після вводу «505»

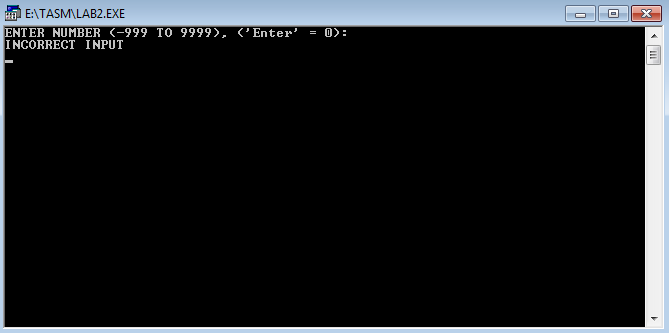
****

Після вводу «-505»****

Ввід «-ten»

****

Вивід помилки

****

**Завдання 2**:

**Текст програми:**

STSEG SEGMENT PARA STACK "STACK"

DB 64 DUP ("STACK")

STSEG ENDS

DSEG SEGMENT PARA PUBLIC "DATA"

CONDITION\_STR DB "z = (19 \* x^3 + 9 \* x^2 - 16 \* x) / (x+1), if 0 < x <= 19", 10, "Z = 1 if x <= 0", 10, "Z = (x^2) / 17, if x > 19", 13, 10, '$'

START\_STR DB "Enter integer number (-999; 255): ", 13, 10, '$' ;задання рядків виведення

ERROR\_STR DB "Incorrect input", 13, 10, '$'

OUTOFRANGE\_STR DB "Number is out of range", 13, 10, '$'

NUM12\_STR DB "Expression can't be calculated with this number (out of range), try 14 > x > 19", 13, 10, '$'

NOINPUT\_STR DB "You haven't entered anything", 13, 10, '$'

OUTPUT\_STR DB "Your result: z = $"

DUMP DB 5, ?, 5 DUP("$")

DIGIT DW 0 ;змінна для збереження чисел

FIRST\_CALC DW 0

SECOND\_CALC DW 0

FIRST\_REMAINDER DW 0

SECOND\_REMAINDER DW 0

NEGATIVE DB 0 ;змінна-флажок для визначення знаку числа

DSEG ENDS

CSEG SEGMENT PARA PUBLIC "CODE"

LOOPM2 MACRO

LOCAL LOOP1, LOOP2

LOOP1:

XOR DX, DX

DIV BX ;ділення АХ на ВХ, остача переходить до DX

ADD DL, 30H

PUSH DX ;збереження значення DX у стеку

INC CX

TEST AX, AX ;перевірка чи є число нулем

JNZ LOOP1

LOOP2:

POP AX ;взяття символу зі стеку

INT 29H ;виведення значення

LOOP LOOP2

ENDM

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

LOOPM1 MACRO

LOCAL LOOP3, LOOP4, LOOP5

LOOP3:

XOR DX, DX

DIV BX ;ділення АХ на ВХ, остача переходить до DX

ADD DL, 30H

PUSH DX ;збереження значення DX у стеку

INC CX

TEST AX, AX ;перевірка чи є число нулем

JNZ LOOP3

LOOP4:

CMP CX, 2

JE LOOP5

MOV AH,2

MOV AL, '0'

INT 29H

LOOP5:

POP AX ;взяття символу зі стеку

INT 29H ;виведення значення

LOOP LOOP5

ENDM

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

FIRST\_CONDITION MACRO ;MACRO якщо х, бiльше 19

MOV BX, DIGIT

MUL BX

MOV FIRST\_CALC, AX

MOV AX, DIGIT

MOV SECOND\_CALC, 17

XOR DX, DX

MOV AX, FIRST\_CALC

MOV BX, SECOND\_CALC

DIV BX

MOV FIRST\_CALC, AX

CMP DX, 0

JNZ REMAINDER\_CALC

JMP CALC\_END

END\_FIRST\_COND2:

RET

ENDM

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

MAIN PROC FAR

ASSUME CS: CSEG, DS: DSEG, SS: STSEG

PUSH DS

PUSH AX

MOV AX,0

MOV AX, DSEG

MOV DS, AX

MOV AH, 9

LEA DX, CONDITION\_STR

INT 21H

CALL NUMBERINPUT ;ввід

CALL CALCULATION

CALL NUMBEROUTPUT ;вивід

MOV AH, 8h ;ввід символу без еха

INT 21h

RET

MAIN ENDP

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

NUMBERINPUT PROC NEAR

MOV AH, 9 ;функція вводу

LEA DX, START\_STR

INT 21H

LEA DX, DUMP ;початкова адреса рядка (зміщення)

MOV AH, 10 ;функція вводу

INT 21H

LEA DI, DUMP+2

MOV AL, [DI] ;[DI] - значення, яке має індекс DI

CMP AL, '-'

JE MINUS

CMP AL, '+'

JE PLUS

CMP AL, 0DH

JE NO\_INPUT

REGISTERCLEAR:

XOR BX, BX ;очищення регістрів та встановлення множника 10

MOV BX, 10

XOR AX, AX

ERRORCHECK:

XOR CX, CX

CMP BYTE PTR [DI], 0DH ;перевірка на кінець рядка

JZ STRINGEND

CMP BYTE PTR [DI], 30H ;перевірка чи є значення не числом зі значенням ASCII менше 0

JB ERROR

CMP BYTE PTR [DI], 39H ;перевірка чи є значення не числом зі значенням ASCII бiльше 9

JA ERROR

MOV CL, [DI]

SUB CL, 30H ;перетворення символу на число

MUL BX

JC ERROR

ADD AX, CX

JC ERROR

INC DI

JMP ERRORCHECK

STRINGEND:

MOV CL, NEGATIVE ;перевірка на знак

CMP CL, 1

JNZ INPUTEND

NEG AX

JMP INPUTEND

PLUS: ;знак плюс

INC DI

JMP REGISTERCLEAR

MINUS:

MOV AL, 1 ;знак мінус

MOV NEGATIVE, AL

INC DI

JMP REGISTERCLEAR

ERROR:

MOV AH, 9 ;виведення повідомлення помилки

LEA DX, ERROR\_STR

INT 21H

MOV AH, 8H ;ввід символу без еха

INT 21H

MOV AH, 4CH ;завершення програми

INT 21H

OUT\_OF\_RANGE:

MOV AH, 9 ;виведення повідомлення помилки

LEA DX, OUTOFRANGE\_STR

INT 21H

MOV AH, 8H ;ввід символу без еха

INT 21H

MOV AH, 4CH ;завершення програми

INT 21H

NUM12:

MOV AH, 9

LEA DX, NUM12\_STR

INT 21H

MOV AH, 8H

INT 21H

MOV AH, 4CH

INT 21H

NO\_INPUT:

MOV AH, 9 ;виведення повідомлення помилки

LEA DX, NOINPUT\_STR

INT 21H

MOV AH, 8H ;ввід символу без еха

INT 21H

MOV AH, 4CH ;завершення програми

INT 21H

INPUTEND:

CMP AX, 255

JG OUT\_OF\_RANGE

CMP AX, 15

JE NUM12

CMP AX, 16

JE NUM12

CMP AX, 17

JE NUM12

CMP AX, 18

JE NUM12

CMP AX, 19

JE NUM12

MOV DIGIT, AX ;присвоєння числа змінній

RET

NUMBERINPUT ENDP

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

CALCULATION PROC NEAR

XOR BX, BX ;очищення регістрів

XOR AX, AX

XOR DX, DX

MOV AX, DIGIT

CMP AX, 0 ;умова, що х <= 0

JLE X\_LOWER

CMP AX, 19 ;умова, що х > 19

JG X\_HIGHER

MOV BX, DIGIT ;умова, що 0 < х <= 19

MOV DX, 19

MUL DX

ADD AX, 9

MUL BX

SUB AX, 16

MUL Bx

MOV FIRST\_CALC, AX

MOV AX, DIGIT

ADD AX, 1

MOV SECOND\_CALC, AX

XOR DX, DX

MOV AX, FIRST\_CALC

MOV BX, SECOND\_CALC

DIV BX

MOV FIRST\_CALC, AX

CMP DX, 0

JNE REMAINDER\_CALC

JMP CALC\_END

X\_LOWER:

MOV FIRST\_CALC, 1

JMP CALC\_END

X\_HIGHER:

FIRST\_CONDITION

JMP CALC\_END

REMAINDER\_CALC: ;розрахунок числа після крапки

XOR CX, CX

REMAINDER\_CONTINUE:

MOV AX, DX

MOV BX, 10

MUL BX

MOV BX, SECOND\_CALC

DIV BX

MOV SECOND\_REMAINDER, DX

INC CX ;послідовне множення остачі на 10 і ділення на задане число до третього знаку

MOV BX, AX

PUSH BX

MOV BX, 10

MOV AX, FIRST\_REMAINDER

MUL BX

MOV DX, SECOND\_REMAINDER

POP BX

ADD AX, BX

MOV FIRST\_REMAINDER, AX

CMP DX, 0

JE CALC\_END

CMP CX, 2

JE CALC\_END

JMP REMAINDER\_CONTINUE

CALC\_END:

RET

CALCULATION ENDP

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

NUMBEROUTPUT PROC NEAR

XOR BX, BX ;очищення регістрів

XOR AX, AX

MOV AH, 9 ;виведення рядка "виводу"

LEA DX, OUTPUT\_STR

INT 21H

MOV BX, FIRST\_CALC ;перенесення значення змінної до ВХ

OR BX, BX

JNS POSITIVECASE1 ;перевірка на знак

MOV AH,2

MOV AL, '-'

INT 29H

MOV BX, FIRST\_REMAINDER

OR BX, BX

JNS NEG\_REMAINDER

NEG BX

MOV FIRST\_REMAINDER, BX

NEG\_REMAINDER:

MOV BX, FIRST\_CALC

NEG BX

JMP POSITIVECASE3

POSITIVECASE1:

CMP BX, 0

JNE POSITIVECASE3

MOV BX, FIRST\_REMAINDER

OR BX, BX

JNS POSITIVECASE2

MOV AH,2

MOV AL, '-'

INT 29H

NEG BX

MOV FIRST\_REMAINDER, BX

POSITIVECASE2:

MOV BX, FIRST\_CALC

POSITIVECASE3:

MOV AX, BX

XOR CX, CX

MOV BX, 10 ;встановлення дільника 10

LOOPM2

MOV AX, FIRST\_REMAINDER

CMP AX, 0

JE OUTPUT\_END

MOV AH,2

MOV AL, '.'

INT 29H

MOV AX, FIRST\_REMAINDER

XOR CX, CX

MOV BX, 10

LOOPM1

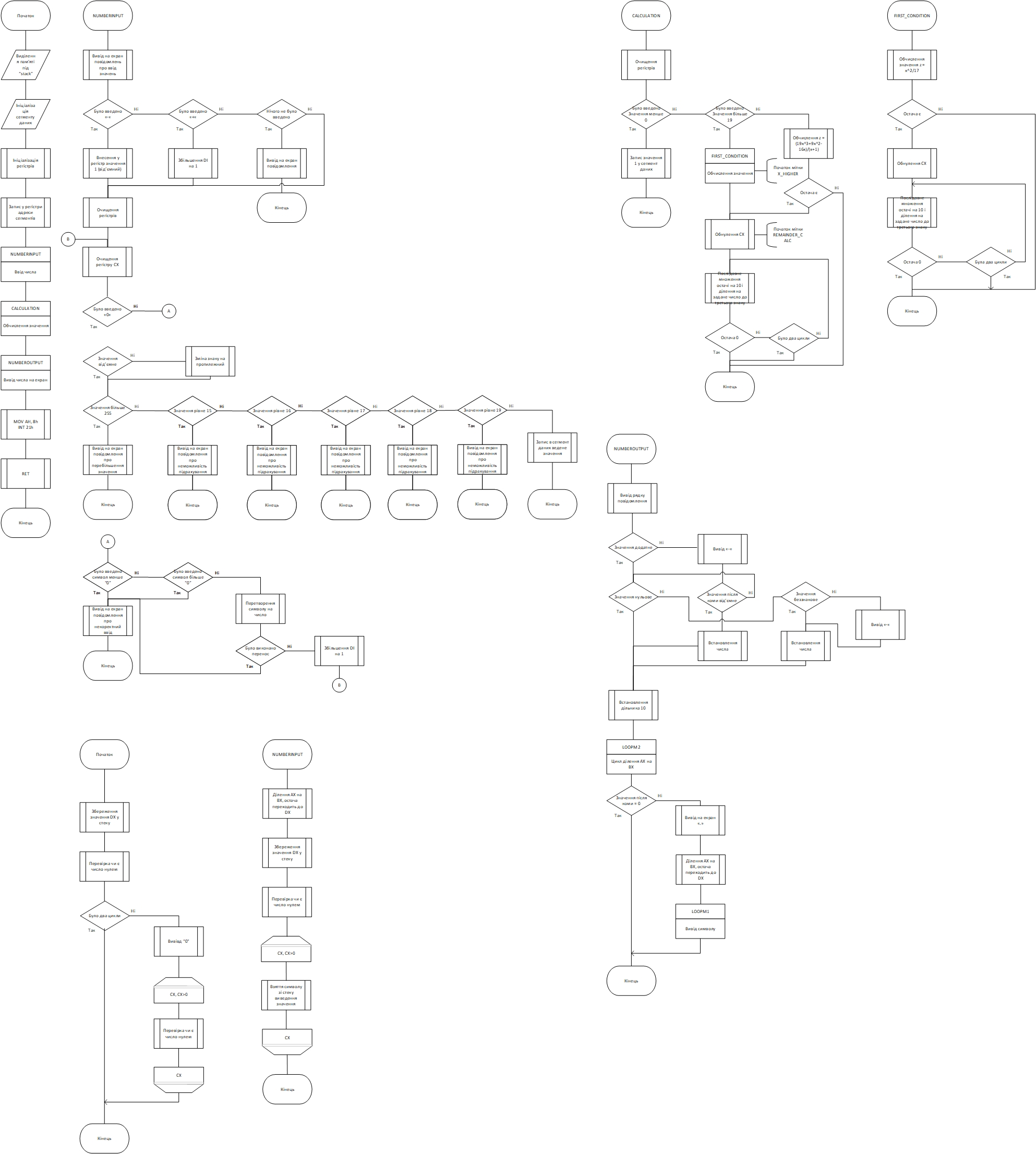
OUTPUT\_END:

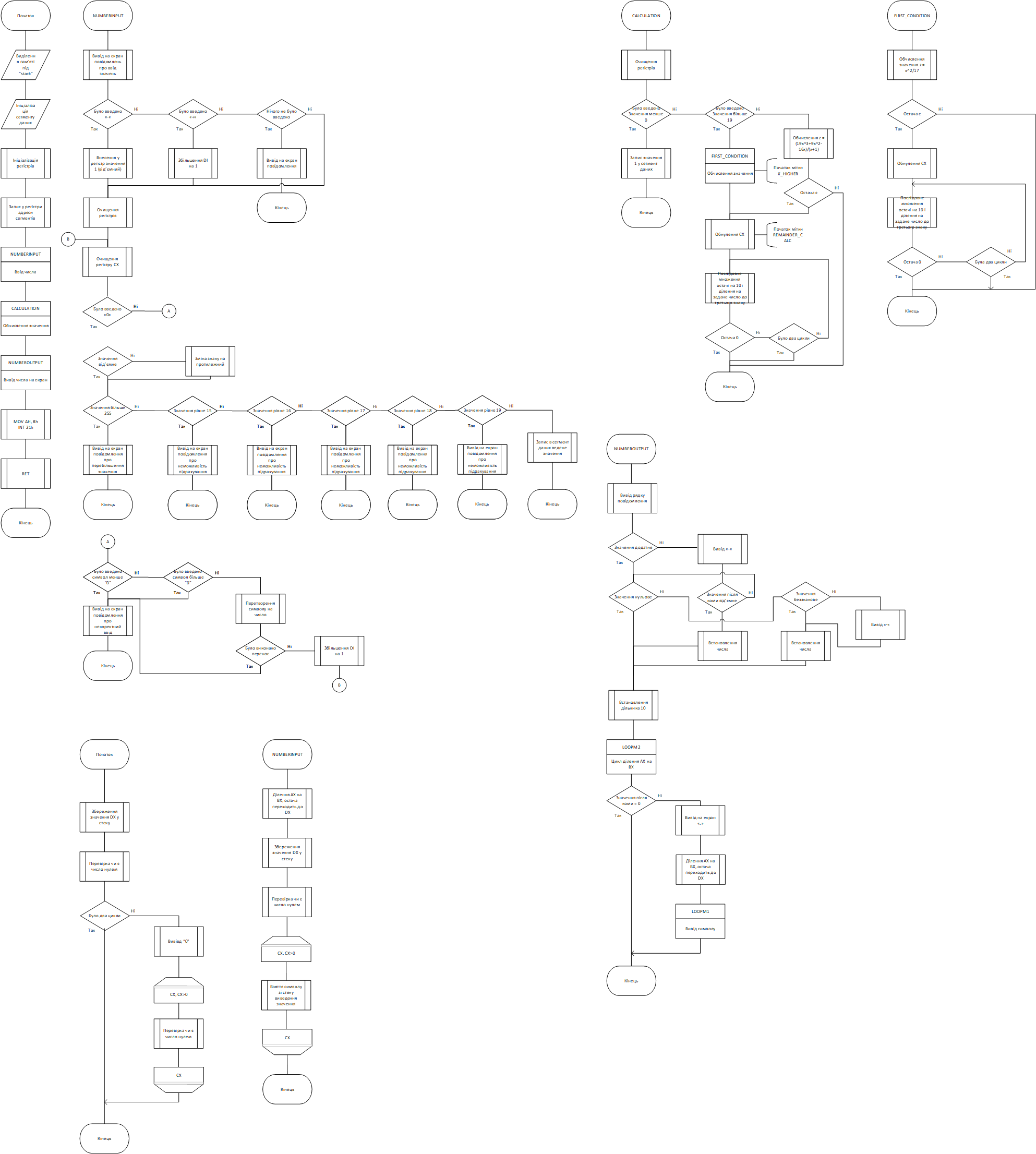
RET

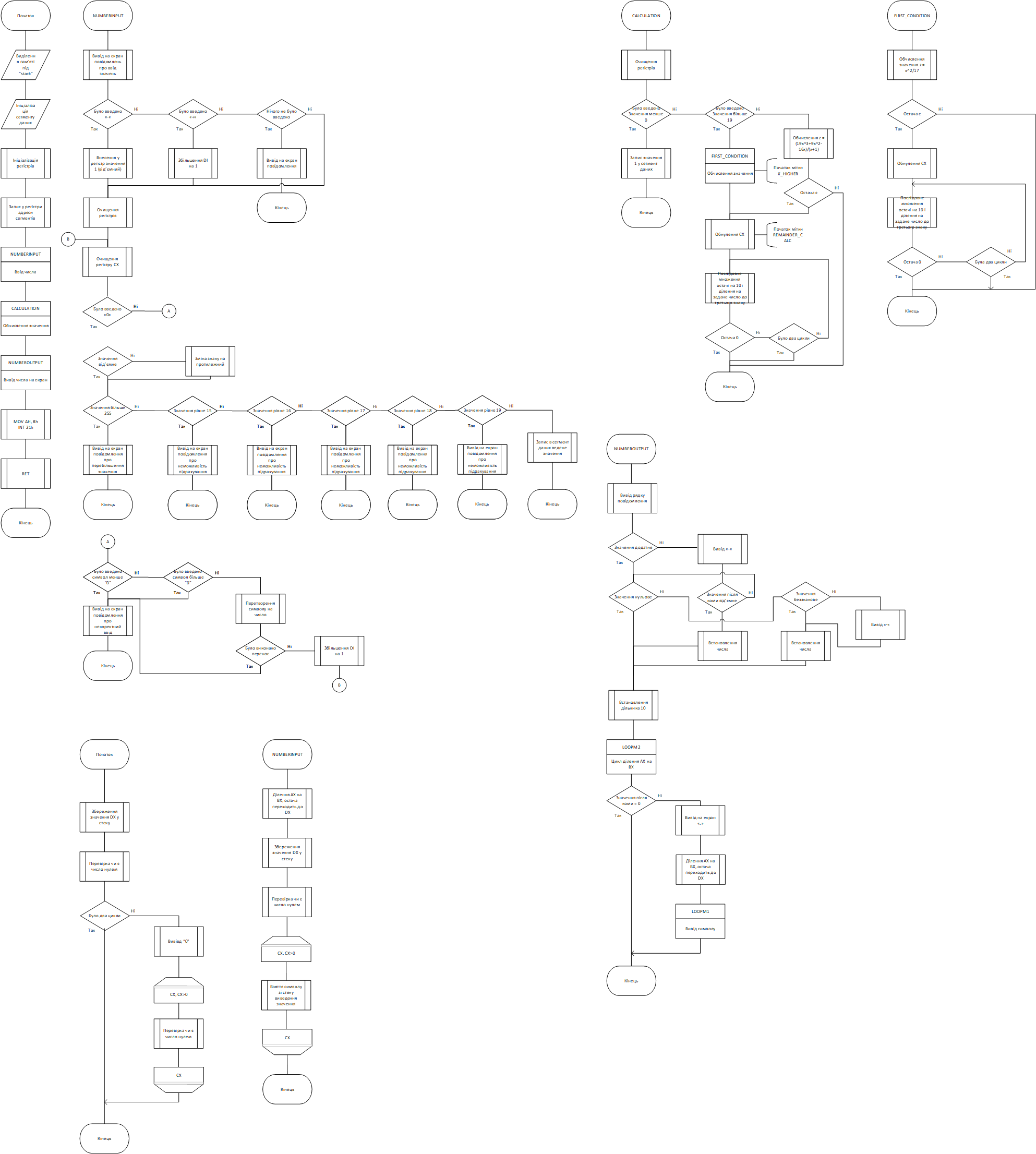
NUMBEROUTPUT ENDP

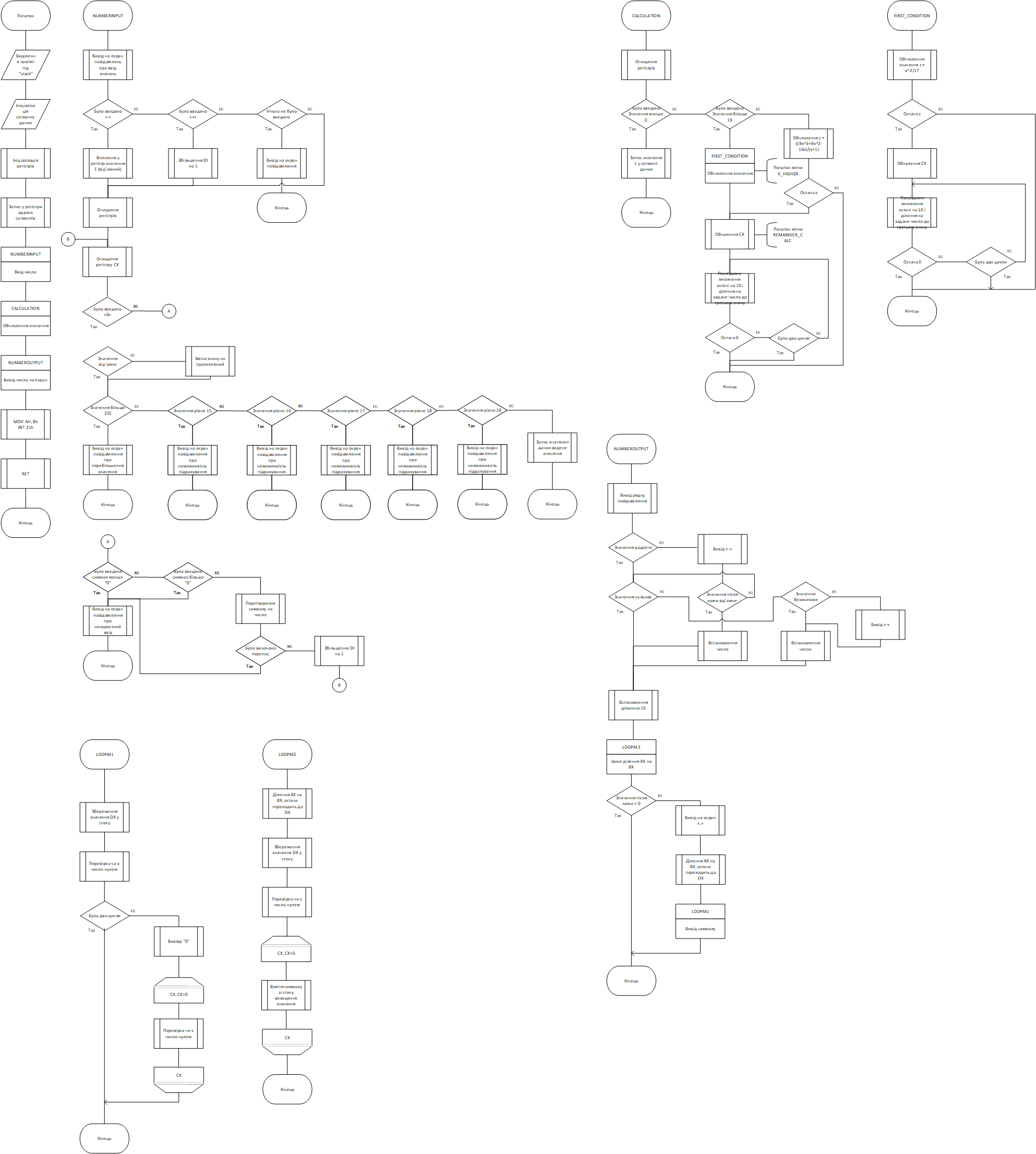
CSEG ENDS

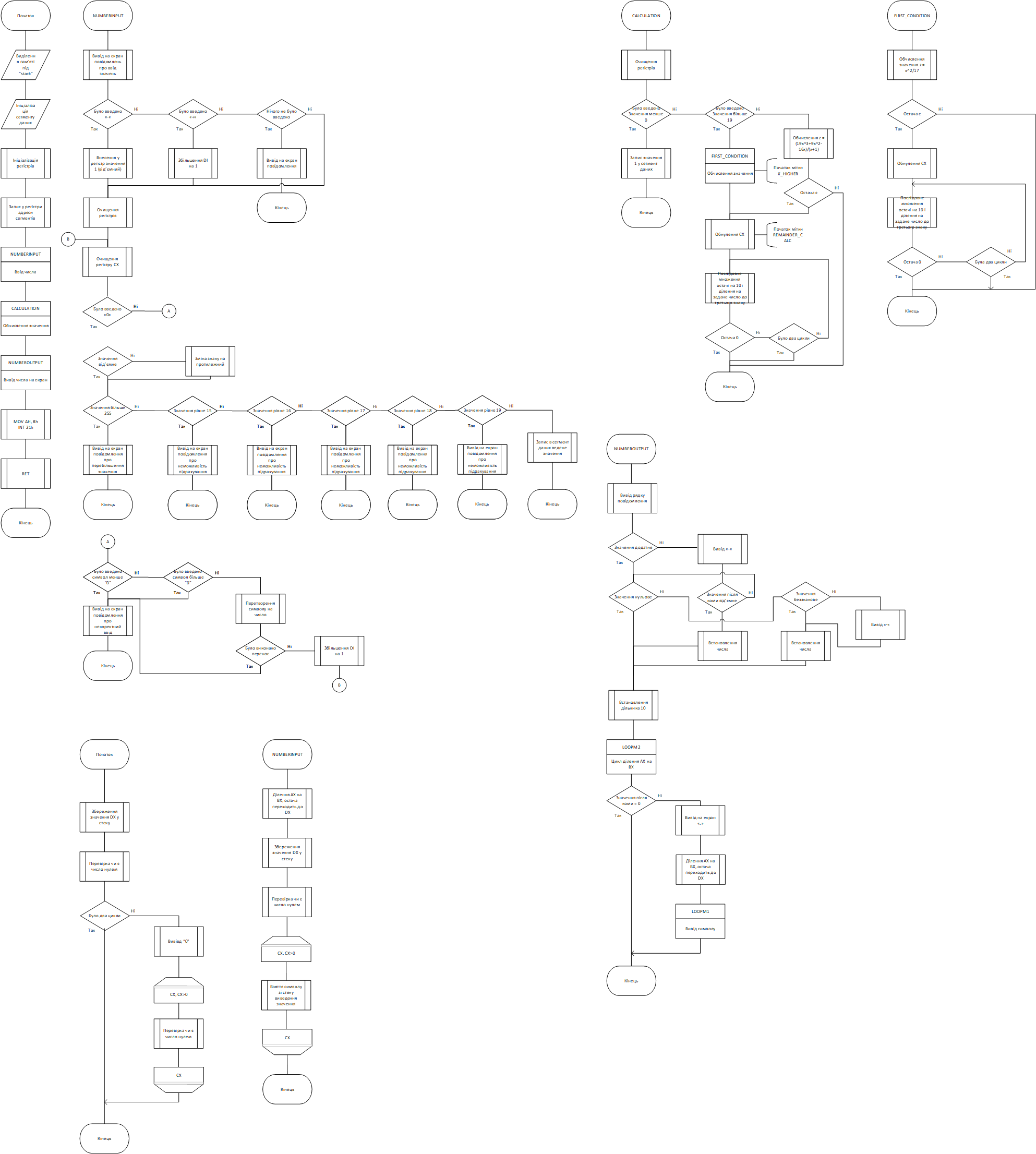
END MAIN

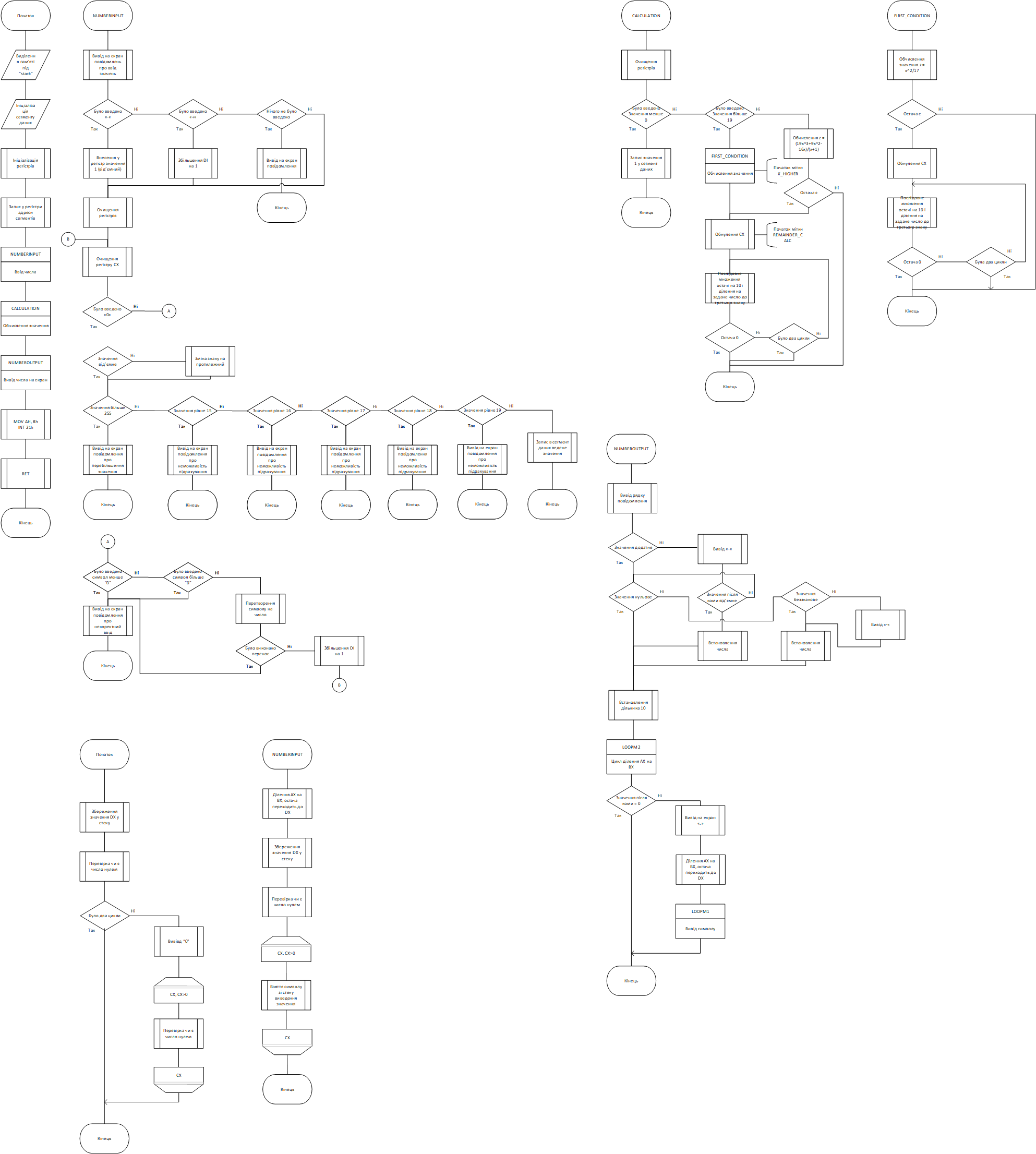
**Схема функціонування програми**



****

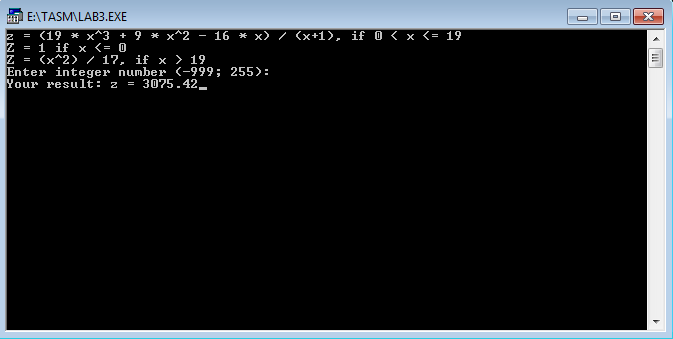
****

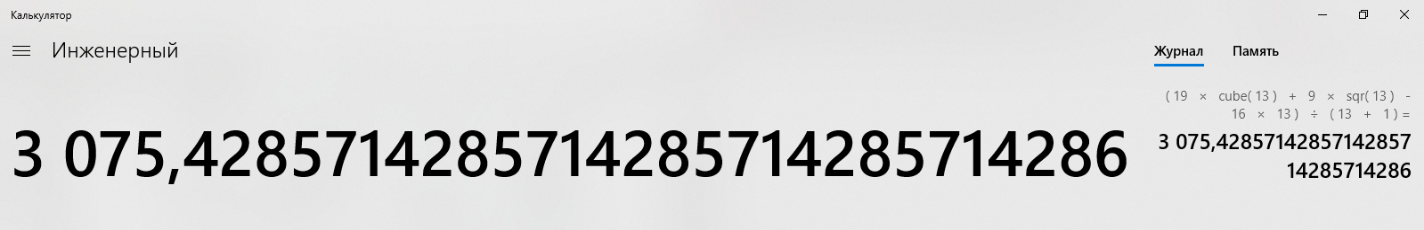
****

****

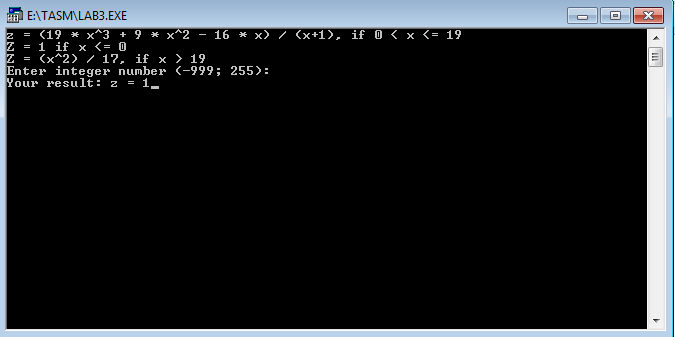
**Вікно роботи програми**

Після вводу «13»

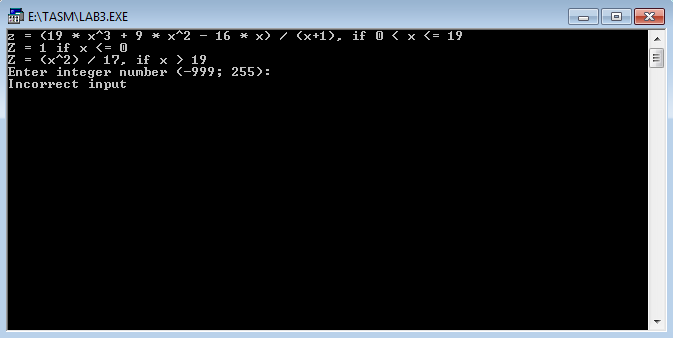




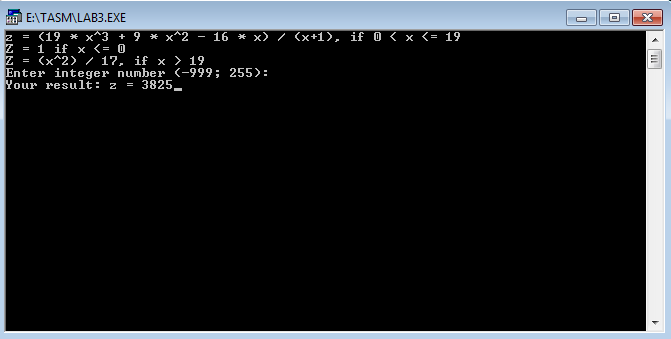
Після вводу «-505»



Ввід «-ten»



Ввід «255»



****

**Завдання 3**

Див. звіт комп’ютерного практикуму №4

**Висновок:**

Було переписано програми 2, 3, 4 з використанням макросів. Програми виконують поставлені задачі коректно. У лістинг файлі відображено «відпрацювання» локальних міток.