

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра технічної кібернетики

Звіти до комп'ютерних практикумів з кредитного модуля «Програмування,  
ч. III» «Системне програмування»

**Прийняв**

**доцент кафедри ТК**

**Лісовиченко О.І.**

**29.12.2020р**

**Виконав**

**Студент групи ІТ-02**

**Макаров І.С.**

## Комп'ютерний практикум №3

**Тема:** програмування розгалужених алгоритмів

**Завдання:**

Завдання: Написати програму, яка буде обчислювати значення функції.

Номер завдання вибирати за останніми двома числами номеру в заліковій книжці.

$$13. \quad Z = \begin{cases} (40x^2 - 23) / x & \text{якщо } 0 < x \leq 7 \\ 38x^3 + 5 & \text{якщо } x \leq 0 \\ 126 / x & \text{якщо } x > 7 \end{cases}$$

**Приклад коду:**

```
.model large
```

```
STK SEGMENT PARA STACK "STACK"  
DB 64 dup ('STACK')  
STK ENDS
```

```
DSEG SEGMENT PARA PUBLIC "DATA"  
digit dw ? ; input X will be stored here
```

```
numerator dw ?  
divider dw ?
```

```
resultFloat dw ?  
resultInt dw ?
```

```
coutFloat dw ?
```

```
askForInput db 'Enter number x: $'  
outOfRangeErrorMessage db 13, 10, 'Number you entered is out of range$'  
nanErrorMessage db 13, 10, 'You entered not a number$'  
calculationErrorMessage db 13, 10, 'Sorry, but I cant perform this calculation ((('  
newLine db " ", 0Dh, 0Ah, '$'
```

```
firstElement db 1 ; first element in row  
isNegative db 0 ; is our number negative
```

```
isError db 0
NUM db 7, ?, 7 dup('?')
```

```
isFirstCondition dw ?
isSecondCondition dw ?
isThirdCondition dw ?
DSEG ENDS
```

```
CSEG SEGMENT PARA PUBLIC "CODE"
ASSUME CS: CSEG, DS: DSEG, SS: STK
```

```
checkFirstCondition PROC
CMP digit, 0 ; check if x greater than 0
JLE endFirstCondition
```

```
CMP digit, 7 ; check if x less than 7
JG endFirstCondition
```

```
MOV isFirstCondition, 1 ;
```

```
MOV BX, digit
MOV AX, 40
CWD
IMUL BX
IMUL BX
```

```
JO overFlowFirst
```

```
SUB AX, 23
MOV numerator, AX ; now numerator =  $40x^2 - 23$ 
```

```
MOV AX, digit
MOV divider, AX ; now divider = x
; and now all we have to do, is to divide numerator on divider
endFirstCondition:
RET
```

```
overFlowFirst:
MOV isFirstCondition, 0
JMP endFirstCondition
checkFirstCondition ENDP
```

```
checkSecondCondition PROC
CMP digit, 0
JG endSecondCondition
```

```
MOV isSecondCondition, 1
```

```
MOV BX, digit
MOV AX, digit
```

```
IMUL BX ; now in AX stored  $x^2$ 
```

```
IMUL BX ; now in AX stored  $x^3$ 
```

```
MOV BX, 38
```

```
IMUL BX ; now in AX stored  $x^3 * 38$ 
```

```
MOV BX, 5
```

```
ADD AX, BX ; adding 5 to  $x^3$ 
```

```
MOV numerator, AX
```

```
MOV divider, 1
```

```
endSecondCondition:
```

```
RET
```

```
overflowSecond:
```

```
MOV isSecondCondition, 0
```

```
JMP endSecondCondition
```

```
checkSecondCondition ENDP
```

```
checkThirdCondition PROC
```

```
CMP digit, 6
```

```
JLE endThirdCondition
```

```
MOV isThirdCondition, 1
```

```
MOV AX, 126
```

```
MOV numerator, AX
```

```
MOV AX, digit
```

```
MOV divider, AX
```

```
endThirdCondition:
```

```
RET
```

```
overflowThird:
```

```
MOV isThirdCondition, 0
```

```
JMP endThirdCondition
```

```
checkThirdCondition ENDP
```

```
divide PROC
```

```
mov AX, numerator ; I can make it better
```

```
cwd
```

```
mov BX, divider
```

```
idiv BX
```

```
mov resultInt, AX
```

```
mov CX, 3
```

```
mov resultFloat, 10
```

```
cmp numerator, 0
```

```
jge a111  
neg dx
```

```
a111:  
mov bx, 10  
mov ax, dx  
mul bx  
jo ercl  
reter:  
cwd  
div divider  
add resultFloat, ax  
cmp cx, 1  
je enddiv
```

```
mov ax, resultFloat  
push dx  
mul bx  
pop dx  
mov resultFloat, ax
```

```
loop a111  
enddiv:  
ret  
ercl:  
mov isError, 1  
JMP enddiv  
divide ENDP
```

```
printNewLine PROC  
LEA DX, newLine  
MOV AH, 09  
int 21h  
ret  
printNewLine ENDP
```

```
CIN PROC  
lea DX, NUM  
XOR AX, AX  
mov ah, 10  
int 21h
```

```
lea SI, NUM + 1  
MOV CL, [SI]
```

```
LEA DI, NUM + 2  
mov DX, 0  
XOR SI, SI  
start:  
MOV BL, [DI]
```

```
A1:
CMP BL, '0'
JB MINUS
CMP BL, '9'
JA MINUS
JMP number
```

```
number:
MOV AX, DX
MOV bX, 10
MUL bX
jo ERROR_OUT_OF_RANGE
MOV DX, AX
XOR AX, AX
MOV AL, [DI]
SUB AX, 30h
ADD DX, AX
jo ERROR_OUT_OF_RANGE
```

```
CMP DX, 32769
JA ERROR_OUT_OF_RANGE
```

```
INC DI
xor BX, BX
mov firstElement, 0
LOOP start
```

```
mov digit, DX
```

```
CMP isNegative, 1
JE negotiate
JNE endcin
```

```
negotiate:
NEG digit
JMP endcin
```

```
MINUS:
XOR AX, AX
CMP firstElement, 1
JNE ERROR_NAN
CMP NUM + 2, '-'
JE MINUSW
JMP ERROR_NAN
ERROR_NAN:
MOV AH, 09
LEA DX, nanErrorMessage
INT 21h
MOV isError, 1
JMP endcin
ERROR_OUT_OF_RANGE:
```

```
MOV AH, 09
LEA DX, outOfRangeErrorMessage
INT 21h
MOV isError, 1
JMP endcin
MINUSW:
MOV isNegative, 1 ; negative
MOV firstElement, 0 ; not first
INC DI
DEC CL
JMP START
endcin:
NEG digit
NEG digit
RET
CIN ENDP
```

```
cout PROC
mov bx, digit
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
cmp coutFloat, 1
je herewego
int 29h
skipcout:
loop m3
jmp endcout
herewego:
mov coutFloat, 0
jmp skipcout
endcout:
ret
cout ENDP
```

```
anotherCout PROC
mov bx, digit
; or bx, bx
jns m12
```

```
mov al, '-'
int 29h
neg bx
```

```
m12:
mov ax, bx
xor cx, cx
mov bx, 10
```

```
m22:
xor dx, dx
div bx
add dl, '0'
push dx
inc cx
test ax, ax
jnz m22
```

```
m32:
pop ax
int 29h
loop m32
```

```
RET
anotherCout ENDP
```

```
MAIN PROC FAR
PUSH DS
MOV AX, 0
PUSH AX
MOV AX, DSEG
MOV DS, AX
```

```
LEA DX, askForInput
MOV AH, 09h
INT 21h
```

```
CALL CIN
CALL printNewLine
```

```
CMP isError, 1
JE endF
```

```
MOV numerator, 1
MOV divider, 1
```



```

CALL checkFirstCondition
CALL checkSecondCondition
CALL checkThirdCondition
CALL divide

CMP isFirstCondition, 1
JE continue
CMP isSecondCondition, 1
JE continue
CMP isThirdCondition, 1
JE continue

JMP calculationError

continue:
XOR AX, AX
MOV AX, resultInt
MOV digit, AX
CALL cout

MOV AL, '.'
INT 29h

MOV AX, resultFloat
MOV digit, AX
MOV coutFloat, 1
CALL cout

endF:
RET
calculationError:
LEA DX, calculationErrorMessage
MOV AH, 09h
INT 21h
MOV isError, 1
JMP endF
MAIN ENDP

CSEG ENDS

END MAIN ; starting the program

```

## Результат роботи програми

```

D:\>laba3.exe
Enter number x: 4
154.250

```

```
D:\>1aba3.exe
```

```
Enter number x: 252
```

```
0.500
```

```
(END)Here is the end of th
```

```
D:\>1aba3.exe
```

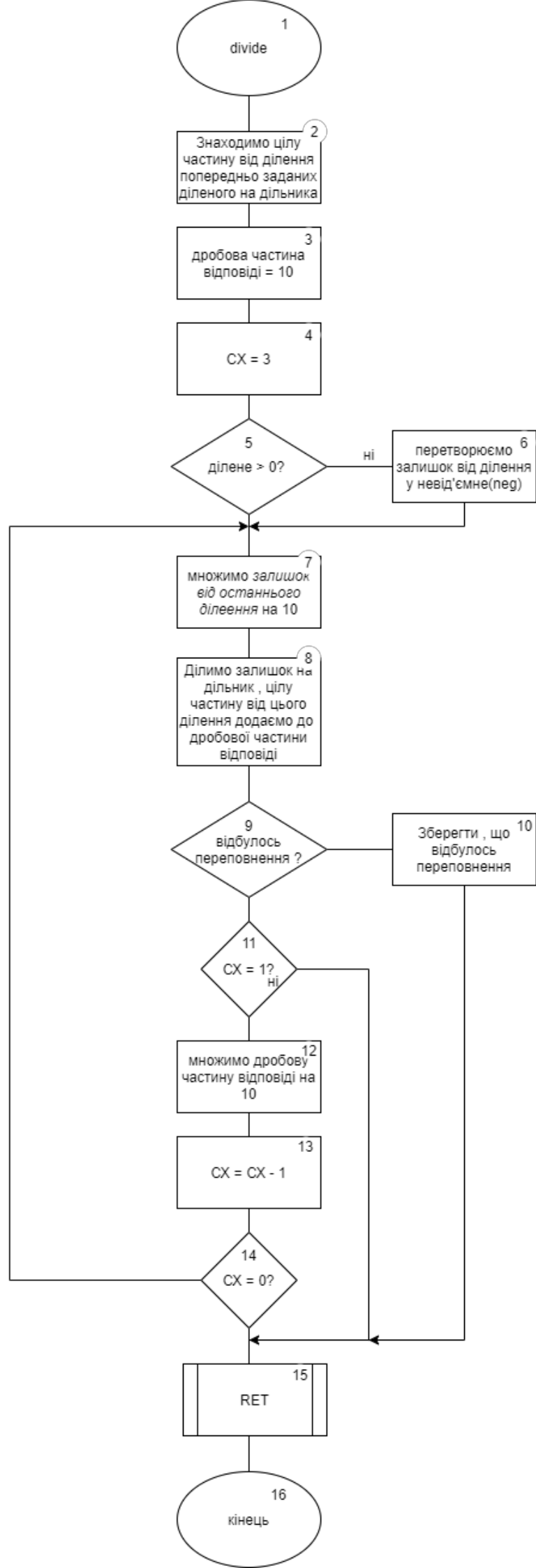
```
Enter number x: -1
```

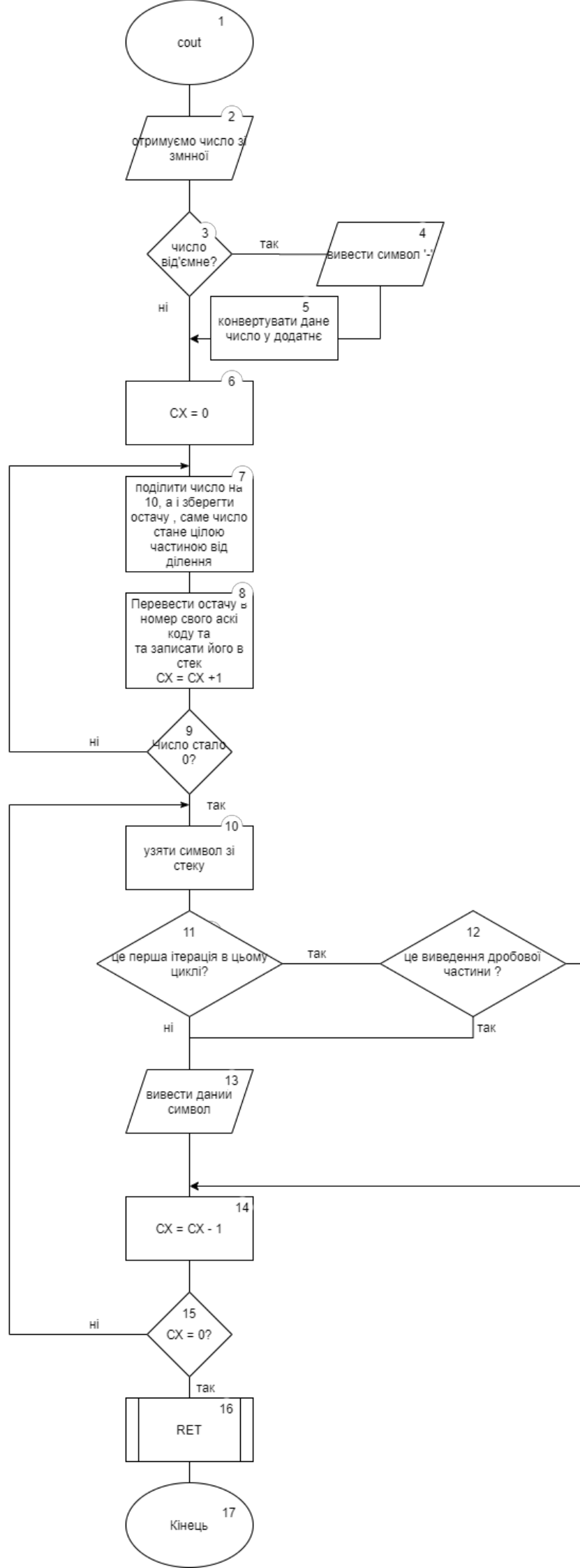
```
-33.000
```

```
(END)Here is the end of
```

## Схема функціонування програми







**Висновок:**

1. Написав програму , що обчислює значення функції

$$13. \quad Z = \begin{cases} (40x^2 - 23) / x & \text{якщо } 0 < x \leq 7 \\ 38x^3 + 5 & \text{якщо } x \leq 0 \\ 126 / x & \text{якщо } x > 7 \end{cases}$$

2. Програма передбачає введення даних , що не зможе обчислити система