**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут ім. Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Лабораторна робота № 8**

з дисципліни

«Системне програмування»

на тему

**«Розробка і використання динамічних бібліотек.»**

Виконав: Перевірив:

Студент 2 курсу ФІОТ доц. Павлов В.Г.

Групи ІМ-13

Ситник Денис Олександрович

Номер у списку групи: 22

Київ – 2023

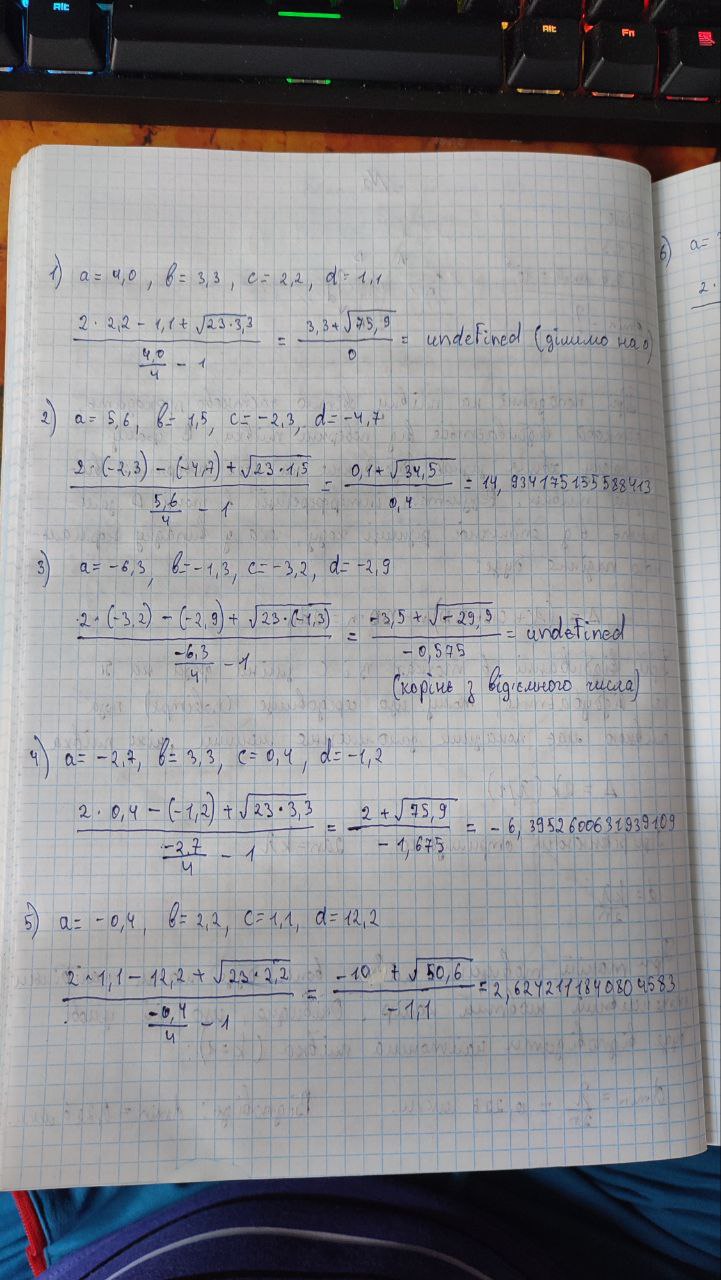
**Мета роботи**

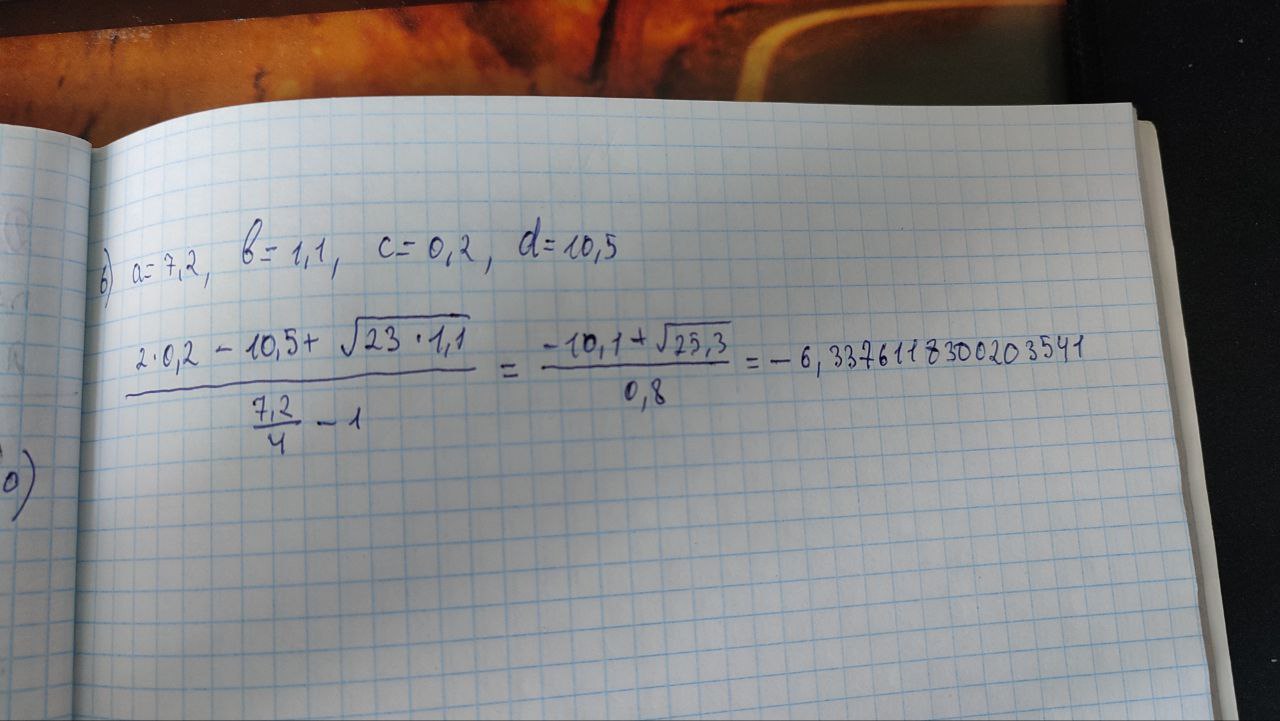
Вивчення прийомів розробки і використання процедур, представлених у вигляді динамічних бібліотек.

Номер у списку – 22, варіант 1:



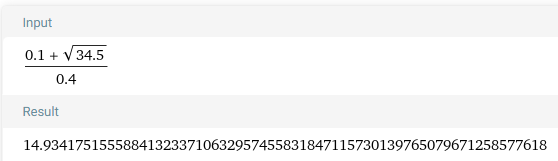
**Контрольні розрахунки:**

****

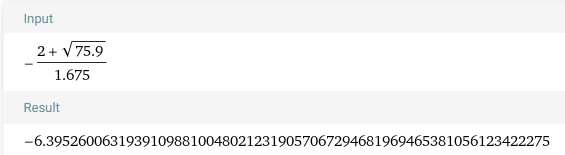
****

**Результати обчислень в інтернет-калькуляторі:**

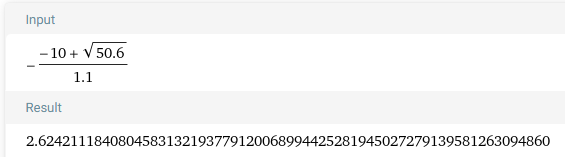
**2)**

****

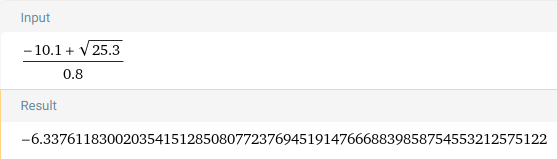
**4)**

****

**5)**

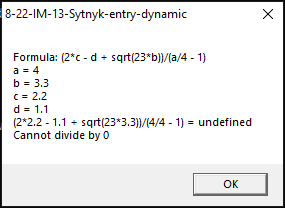
****

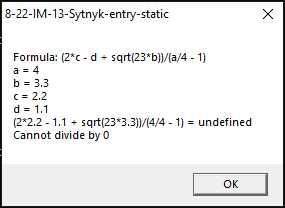
**6)**

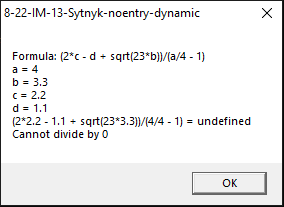
****

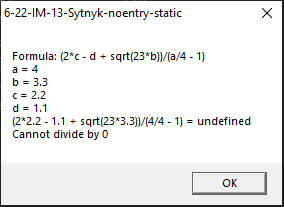
**Скріншоти виконання програм:**

Нуль у знаменнику:

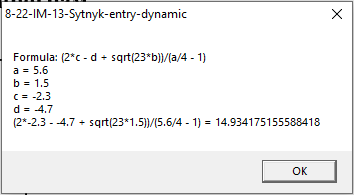


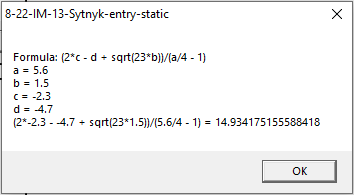


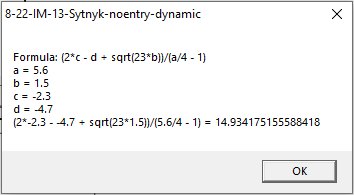


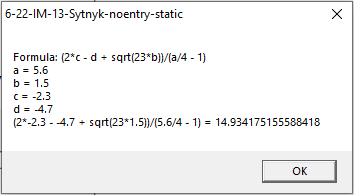


Додатнє на додатнє:



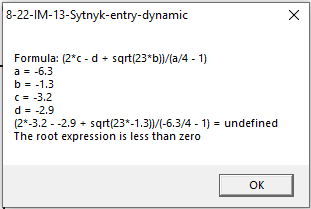


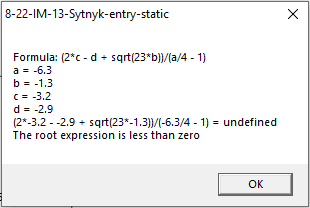


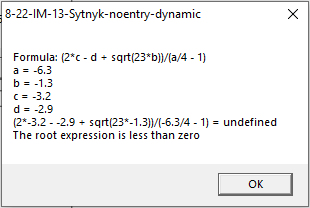


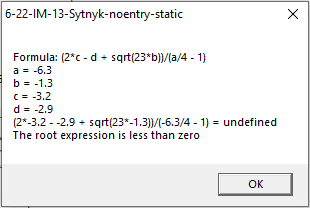
14,934175155588413 у контрольних розрахунках (відрізняється остання цифра)

Від’ємний підкореневий вираз:

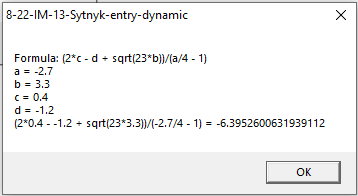


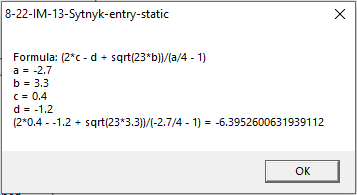


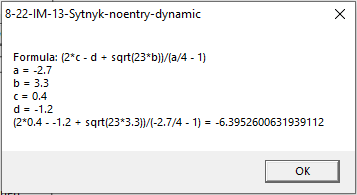


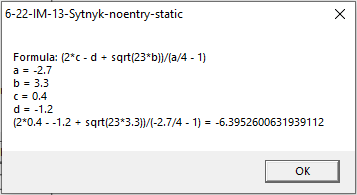


Додатнє на від’ємне:



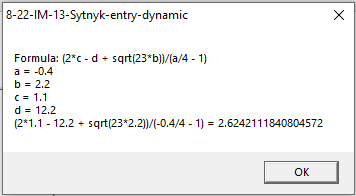


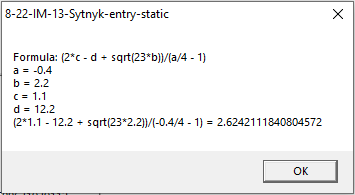


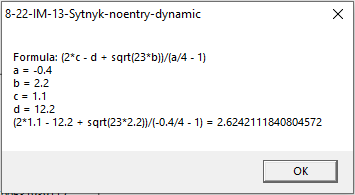


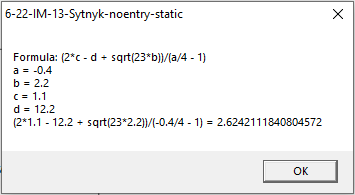
-6,3952600631939109 у контрольних розрахунках (відрізняються дві останні цифри)

Від’ємне на від‘ємне:



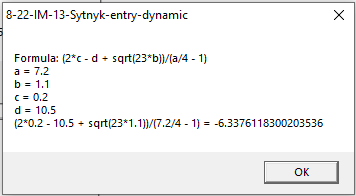


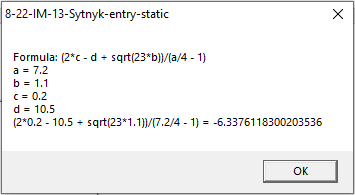


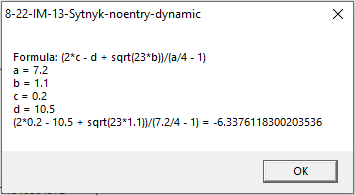


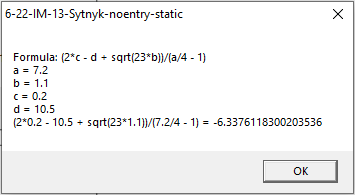
2,6242111840804583 у контрольних розрахунках (відрізняються дві останні цифри)

Від’ємне на додатнє









-6,3376118300203541 у контрольних розрахунках (відрізняються дві останні цифри)

**Ці відмінності пояснюються тим, що комп’ютер округлює числа при розрахунках.**

**Лістинги:**

**Динамічна із точкою входу**

**Основний файл**

.386

.model flat, stdcall

option casemap :none

include \masm32\include\masm32rt.inc

includelib \masm32\lib\kernel32.lib

mov\_constant macro const

mov edx, offset const

push edx

endm

lea\_value macro value

lea edx, value

push edx

endm

push\_into\_stack macro

mov\_constant result

mov\_constant twentyThree

mov\_constant four

mov\_constant two

mov\_constant one

lea\_value valuesD[ebp \* 8]

lea\_value valuesC[ebp \* 8]

lea\_value valuesB[ebp \* 8]

lea\_value valuesA[ebp \* 8]

endm

coprocessorState macro

ftst

fnstsw ax

sahf

endm

successMessage macro

invoke wsprintf, addr buff, addr dataFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA, addr resultStr

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

errorMessage macro messageFormat

invoke wsprintf, addr buff, addr messageFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

.data?

library\_loader dd ?

procedure\_pointer dd ?

result dq 32 dup(?)

resultStr db 32 dup(?)

valueA db 32 dup(?)

valueB db 32 dup(?)

valueC db 32 dup(?)

valueD db 32 dup(?)

data db 32 dup(?)

buff db 32 dup(?)

.data

procedure\_name db "calculateLib", 0

library\_name db "8-22-IM-13-Sytnyk-lib", 0

labTitle db "8-22-IM-13-Sytnyk-entry-dynamic", 0

dataFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = %s", 10, 0

zeroDenominatorFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"Cannot divide by 0", 10, 0

invalidDefinitionAreaFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"The root expression is less than zero", 10, 0

valuesA dq 4.0, 5.6, -6.3, -2.7, -0.4, 7.2

valuesB dq 3.3, 1.5, -1.3, 3.3, 2.2, 1.1

valuesC dq 2.2, -2.3, -3.2, 0.4, 1.1, 0.2

valuesD dq 1.1, -4.7, -2.9, -1.2, 12.2, 10.5

one dq 1.0

two dq 2.0

four dq 4.0

twentyThree dq 23.0

.code

main:

mov ebp, 0

invoke LoadLibrary, addr library\_name

mov library\_loader, eax

invoke GetProcAddress, library\_loader, addr procedure\_name

mov procedure\_pointer, eax

.while ebp < 6

;; Converting numbers to strings

invoke FloatToStr2, valuesA[ebp \* 8], addr valueA

invoke FloatToStr2, valuesB[ebp \* 8], addr valueB

invoke FloatToStr2, valuesC[ebp \* 8], addr valueC

invoke FloatToStr2, valuesD[ebp \* 8], addr valueD

push\_into\_stack

;; Checking denominator

finit

fld four ;; 4 in ctack

fld valuesA[ebp \* 8] ;; a in ctack

fdiv st(0), st(1) ;; a/4

fld one ;; 1 in stack

fxch st(1) ;; change positions

fsub st(0), st(1) ;; a/4 - 1

coprocessorState

jz zeroDenominator

;; Checking square root

fld twentyThree ;; 23 in stack

fmul valuesB[ebp \* 8] ;; 23\*b

fsqrt ;; sqrt(23\*b)

coprocessorState

test ah, 01000000b

jnz invalidDefinitionArea

call procedure\_pointer

invoke FloatToStr2, result, addr resultStr ;; converting result into string

successMessage

jmp next

zeroDenominator:

errorMessage zeroDenominatorFormat

jmp next

invalidDefinitionArea:

errorMessage invalidDefinitionAreaFormat

jmp next

next:

inc ebp

mov data, 0h

.endw

invoke ExitProcess, 0

end main

**Файл з бібліотекою**

.386

.model flat, stdcall

option casemap :none

.code

calculateLib\_entry proc hInstDLL: dword, reason: dword, unuser: dword

mov eax, 1

ret

calculateLib\_entry endp

calculateLib proc valuesA: ptr qword, valuesB: ptr qword, valuesC: ptr qword,

valuesD: ptr qword, one: ptr qword, two: ptr qword,

four: ptr qword, twentyThree: ptr qword, result: ptr qword

finit

mov ebx, twentyThree

fld qword ptr [ebx] ;; 23 in stack

mov ebx, valuesB

fmul qword ptr [ebx] ;; 23\*b

fsqrt ;; sqrt(23\*b)

mov ebx, two

fld qword ptr [ebx] ;; 2 in stack

mov ebx, valuesC

fmul qword ptr [ebx] ;; 2\*c

mov ebx, valuesD

fsub qword ptr [ebx] ;; 2\*c - d

fadd ;; numerator calculated

mov ebx, valuesA

fld qword ptr [ebx] ;; a in stack

mov ebx, four

fdiv qword ptr [ebx] ;; a/4

mov ebx, one

fsub qword ptr [ebx] ;; a/4 - 1

fxch st(1) ;; change positions

fdiv st(0), st(1) ;; numerator/denominator

mov ebx, result

fstp qword ptr [ebx]

ret

calculateLib endp

end calculateLib\_entry

**Файл .bat**

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk-lib.asm"

\masm32\bin\Link.exe /out:"8-22-IM-13-Sytnyk-lib.dll" /export:calculateLib /dll "8-22-IM-13-Sytnyk-lib.obj"

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk.asm"

\masm32\bin\Link.exe /subsystem:windows "8-22-IM-13-Sytnyk.obj"

8-22-IM-13-Sytnyk.exe

**Статична з точкою входу**

**Основний файл**

.386

.model flat, stdcall

option casemap :none

include \masm32\include\masm32rt.inc

includelib 8-22-IM-13-Sytnyk-lib.lib

calculateLib proto :ptr qword, :ptr qword, :ptr qword, :ptr qword, :ptr qword, :ptr qword, :ptr qword, :ptr qword, :ptr qword

coprocessorState macro

ftst

fnstsw ax

sahf

endm

successMessage macro

invoke wsprintf, addr buff, addr dataFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA, addr resultStr

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

errorMessage macro messageFormat

invoke wsprintf, addr buff, addr messageFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

.data?

result dq 32 dup(?)

resultStr db 32 dup(?)

valueA db 32 dup(?)

valueB db 32 dup(?)

valueC db 32 dup(?)

valueD db 32 dup(?)

data db 32 dup(?)

buff db 32 dup(?)

.data

labTitle db "8-22-IM-13-Sytnyk-entry-static", 0

dataFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = %s", 10, 0

zeroDenominatorFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"Cannot divide by 0", 10, 0

invalidDefinitionAreaFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"The root expression is less than zero", 10, 0

valuesA dq 4.0, 5.6, -6.3, -2.7, -0.4, 7.2

valuesB dq 3.3, 1.5, -1.3, 3.3, 2.2, 1.1

valuesC dq 2.2, -2.3, -3.2, 0.4, 1.1, 0.2

valuesD dq 1.1, -4.7, -2.9, -1.2, 12.2, 10.5

one dq 1.0

two dq 2.0

four dq 4.0

twentyThree dq 23.0

.code

main:

mov ebp, 0

.while ebp < 6

;; Converting numbers to strings

invoke FloatToStr2, valuesA[ebp \* 8], addr valueA

invoke FloatToStr2, valuesB[ebp \* 8], addr valueB

invoke FloatToStr2, valuesC[ebp \* 8], addr valueC

invoke FloatToStr2, valuesD[ebp \* 8], addr valueD

invoke calculateLib, addr valuesA[ebp \* 8], addr valuesB[ebp \* 8], addr valuesC[ebp \* 8], addr valuesD[ebp \* 8],

addr one, addr two, addr four, addr twentyThree, addr result

;; Checking denominator

fld four ;; 4 in ctack

fld valuesA[ebp \* 8] ;; a in ctack

fdiv st(0), st(1) ;; a/4

fld one ;; 1 in stack

fxch st(1) ;; change positions

fsub st(0), st(1) ;; a/4 - 1

coprocessorState

jz zeroDenominator

;; Checking square root

fld twentyThree ;; 23 in stack

fmul valuesB[ebp \* 8] ;; 23\*b

fsqrt ;; sqrt(23\*b)

coprocessorState

test ah, 01000000b

jnz invalidDefinitionArea

invoke FloatToStr2, result, addr resultStr ;; converting result into string

successMessage

jmp next

zeroDenominator:

errorMessage zeroDenominatorFormat

jmp next

invalidDefinitionArea:

errorMessage invalidDefinitionAreaFormat

jmp next

next:

inc ebp

mov data, 0h

.endw

invoke ExitProcess, 0

end main

**Файл з бібліотекою**

.386

.model flat, stdcall

option casemap :none

.code

calculateLib\_entry proc hInstDLL: dword, reason: dword, unuser: dword

mov eax, 1

ret

calculateLib\_entry endp

calculateLib proc valuesA: ptr qword, valuesB: ptr qword, valuesC: ptr qword,

valuesD: ptr qword, one: ptr qword, two: ptr qword,

four: ptr qword, twentyThree: ptr qword, result: ptr qword

finit

mov ebx, twentyThree

fld qword ptr [ebx] ;; 23 in stack

mov ebx, valuesB

fmul qword ptr [ebx] ;; 23\*b

fsqrt ;; sqrt(23\*b)

mov ebx, two

fld qword ptr [ebx] ;; 2 in stack

mov ebx, valuesC

fmul qword ptr [ebx] ;; 2\*c

mov ebx, valuesD

fsub qword ptr [ebx] ;; 2\*c - d

fadd ;; numerator calculated

mov ebx, valuesA

fld qword ptr [ebx] ;; a in stack

mov ebx, four

fdiv qword ptr [ebx] ;; a/4

mov ebx, one

fsub qword ptr [ebx] ;; a/4 - 1

fxch st(1) ;; change positions

fdiv st(0), st(1) ;; numerator/denominator

mov ebx, result

fstp qword ptr [ebx]

ret

calculateLib endp

end calculateLib\_entry

**Файл .bat**

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk-lib.asm"

\masm32\bin\Link.exe /out:"8-22-IM-13-Sytnyk-lib.dll" /export:calculateLib /dll "8-22-IM-13-Sytnyk-lib.obj"

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk.asm"

\masm32\bin\Link.exe /subsystem:windows "8-22-IM-13-Sytnyk.obj"

8-22-IM-13-Sytnyk.exe

**Файл .def**

LIBRARY 8-22-IM-13-Sytnyk-lib

EXPORTS calculateLib

**Динамічна без точки входу**

**Основний файл**

.386

.model flat, stdcall

option casemap :none

include \masm32\include\masm32rt.inc

includelib \masm32\lib\kernel32.lib

mov\_constant macro const

mov edx, offset const

push edx

endm

lea\_value macro value

lea edx, value

push edx

endm

push\_to\_stack macro

mov\_constant result

mov\_constant twentyThree

mov\_constant four

mov\_constant two

mov\_constant one

lea\_value valuesD[ebp \* 8]

lea\_value valuesC[ebp \* 8]

lea\_value valuesB[ebp \* 8]

lea\_value valuesA[ebp \* 8]

endm

coprocessorState macro

ftst

fnstsw ax

sahf

endm

successMessage macro

invoke wsprintf, addr buff, addr dataFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA, addr resultStr

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

errorMessage macro messageFormat

invoke wsprintf, addr buff, addr messageFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

.data?

library\_loader dd ?

procedure\_pointer dd ?

result dq 32 dup(?)

resultStr db 32 dup(?)

valueA db 32 dup(?)

valueB db 32 dup(?)

valueC db 32 dup(?)

valueD db 32 dup(?)

data db 32 dup(?)

buff db 32 dup(?)

.data

procedure\_name db "calculateLib", 0

library\_name db "8-22-IM-13-Sytnyk-lib", 0

labTitle db "6-22-IM-13-Sytnyk", 0

dataFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = %s", 10, 0

zeroDenominatorFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"Cannot divide by 0", 10, 0

invalidDefinitionAreaFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"The root expression is less than zero", 10, 0

valuesA dq 4.0, 5.6, -6.3, -2.7, -0.4, 7.2

valuesB dq 3.3, 1.5, -1.3, 3.3, 2.2, 1.1

valuesC dq 2.2, -2.3, -3.2, 0.4, 1.1, 0.2

valuesD dq 1.1, -4.7, -2.9, -1.2, 12.2, 10.5

one dq 1.0

two dq 2.0

four dq 4.0

twentyThree dq 23.0

.code

main:

mov ebp, 0

invoke LoadLibrary, addr library\_name

mov library\_loader, eax

invoke GetProcAddress, library\_loader, addr procedure\_name

mov procedure\_pointer, eax

.while ebp < 6

;; Converting numbers to strings

invoke FloatToStr2, valuesA[ebp \* 8], addr valueA

invoke FloatToStr2, valuesB[ebp \* 8], addr valueB

invoke FloatToStr2, valuesC[ebp \* 8], addr valueC

invoke FloatToStr2, valuesD[ebp \* 8], addr valueD

prepare\_to\_work

;; Checking denominator

finit

fld four ;; 4 in ctack

fld valuesA[ebp \* 8] ;; a in ctack

fdiv st(0), st(1) ;; a/4

fld one ;; 1 in stack

fxch st(1) ;; change positions

fsub st(0), st(1) ;; a/4 - 1

coprocessorState

jz zeroDenominator

;; Checking square root

fld twentyThree ;; 23 in stack

fmul valuesB[ebp \* 8] ;; 23\*b

fsqrt ;; sqrt(23\*b)

coprocessorState

test ah, 01000000b

jnz invalidDefinitionArea

call push\_to\_stack

invoke FloatToStr2, result, addr resultStr ;; converting result into string

successMessage

jmp next

zeroDenominator:

errorMessage zeroDenominatorFormat

jmp next

invalidDefinitionArea:

errorMessage invalidDefinitionAreaFormat

jmp next

next:

inc ebp

mov data, 0h

.endw

invoke ExitProcess, 0

end main

**Файл з бібліотекою**

.386

.model flat, stdcall

option casemap :none

.code

calculateLib proc valuesA: ptr qword, valuesB: ptr qword, valuesC: ptr qword,

valuesD: ptr qword, one: ptr qword, two: ptr qword,

four: ptr qword, twentyThree: ptr qword, result: ptr qword

finit

mov ebx, twentyThree

fld qword ptr [ebx] ;; 23 in stack

mov ebx, valuesB

fmul qword ptr [ebx] ;; 23\*b

fsqrt ;; sqrt(23\*b)

mov ebx, two

fld qword ptr [ebx] ;; 2 in stack

mov ebx, valuesC

fmul qword ptr [ebx] ;; 2\*c

mov ebx, valuesD

fsub qword ptr [ebx] ;; 2\*c - d

fadd ;; numerator calculated

mov ebx, valuesA

fld qword ptr [ebx] ;; a in stack

mov ebx, four

fdiv qword ptr [ebx] ;; a/4

mov ebx, one

fsub qword ptr [ebx] ;; a/4 - 1

fxch st(1) ;; change positions

fdiv st(0), st(1) ;; numerator/denominator

mov ebx, result

fstp qword ptr [ebx]

ret

calculateLib endp

end

**Файл .bat**

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk-lib.asm"

\masm32\bin\Link.exe /out:"8-22-IM-13-Sytnyk-lib.dll" /export:calculateLib /dll /noentry "8-22-IM-13-Sytnyk-lib.obj"

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk.asm"

\masm32\bin\Link.exe /subsystem:windows "8-22-IM-13-Sytnyk.obj"

8-22-IM-13-Sytnyk.exe

**Статична без точки входу**

**Основний файл**

.386

.model flat, stdcall

option casemap :none

include \masm32\include\masm32rt.inc

includelib 8-22-IM-13-Sytnyk-lib.lib

calculateLib proto :ptr qword, :ptr qword, :ptr qword, :ptr qword,

:ptr qword, :ptr qword, :ptr qword, :ptr qword, :ptr qword

coprocessorState macro

ftst

fnstsw ax

sahf

endm

successMessage macro

invoke wsprintf, addr buff, addr dataFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA, addr resultStr

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

errorMessage macro messageFormat

invoke wsprintf, addr buff, addr messageFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

.data?

result dq 32 dup(?)

resultStr db 32 dup(?)

valueA db 32 dup(?)

valueB db 32 dup(?)

valueC db 32 dup(?)

valueD db 32 dup(?)

data db 32 dup(?)

buff db 32 dup(?)

.data

labTitle db "6-22-IM-13-Sytnyk-noentry-static", 0

dataFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = %s", 10, 0

zeroDenominatorFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"Cannot divide by 0", 10, 0

invalidDefinitionAreaFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"The root expression is less than zero", 10, 0

valuesA dq 4.0, 5.6, -6.3, -2.7, -0.4, 7.2

valuesB dq 3.3, 1.5, -1.3, 3.3, 2.2, 1.1

valuesC dq 2.2, -2.3, -3.2, 0.4, 1.1, 0.2

valuesD dq 1.1, -4.7, -2.9, -1.2, 12.2, 10.5

one dq 1.0

two dq 2.0

four dq 4.0

twentyThree dq 23.0

.code

main:

mov ebp, 0

.while ebp < 6

;; Converting numbers to strings

invoke FloatToStr2, valuesA[ebp \* 8], addr valueA

invoke FloatToStr2, valuesB[ebp \* 8], addr valueB

invoke FloatToStr2, valuesC[ebp \* 8], addr valueC

invoke FloatToStr2, valuesD[ebp \* 8], addr valueD

invoke calculateLib, addr valuesA[ebp \* 8], addr valuesB[ebp \* 8], addr valuesC[ebp \* 8], addr valuesD[ebp \* 8],

addr one, addr two, addr four, addr twentyThree, addr result

;; Checking denominator

fld four ;; 4 in ctack

fld valuesA[ebp \* 8] ;; a in ctack

fdiv st(0), st(1) ;; a/4

fld one ;; 1 in stack

fxch st(1) ;; change positions

fsub st(0), st(1) ;; a/4 - 1

coprocessorState

jz zeroDenominator

;; Checking square root

fld twentyThree ;; 23 in stack

fmul valuesB[ebp \* 8] ;; 23\*b

fsqrt ;; sqrt(23\*b)

coprocessorState

test ah, 01000000b

jnz invalidDefinitionArea

invoke FloatToStr2, result, addr resultStr ;; converting result into string

successMessage

jmp next

zeroDenominator:

errorMessage zeroDenominatorFormat

jmp next

invalidDefinitionArea:

errorMessage invalidDefinitionAreaFormat

jmp next

next:

inc ebp

mov data, 0h

.endw

invoke ExitProcess, 0

end main

**Файл з бібліотекою**

.386

.model flat, stdcall

option casemap :none

.code

calculateLib proc valuesA: ptr qword, valuesB: ptr qword, valuesC: ptr qword,

valuesD: ptr qword, one: ptr qword, two: ptr qword,

four: ptr qword, twentyThree: ptr qword, result: ptr qword

finit

mov ebx, twentyThree

fld qword ptr [ebx] ;; 23 in stack

mov ebx, valuesB

fmul qword ptr [ebx] ;; 23\*b

fsqrt ;; sqrt(23\*b)

mov ebx, two

fld qword ptr [ebx] ;; 2 in stack

mov ebx, valuesC

fmul qword ptr [ebx] ;; 2\*c

mov ebx, valuesD

fsub qword ptr [ebx] ;; 2\*c - d

fadd ;; numerator calculated

mov ebx, valuesA

fld qword ptr [ebx] ;; a in stack

mov ebx, four

fdiv qword ptr [ebx] ;; a/4

mov ebx, one

fsub qword ptr [ebx] ;; a/4 - 1

fxch st(1) ;; change positions

fdiv st(0), st(1) ;; numerator/denominator

mov ebx, result

fstp qword ptr [ebx]

ret

calculateLib endp

end

**Файл .bat**

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk-lib.asm"

\masm32\bin\Link.exe /out:"8-22-IM-13-Sytnyk-lib.dll" /def:"8-22-IM-13-Sytnyk.def" /dll /noentry "8-22-IM-13-Sytnyk-lib.obj"

\masm32\bin\ml /c /coff "8-22-IM-13-Sytnyk.asm"

\masm32\bin\Link.exe /subsystem:windows "8-22-IM-13-Sytnyk.obj"

8-22-IM-13-Sytnyk.exe

**Файл .def**

LIBRARY 8-22-IM-13-Sytnyk-lib

EXPORTS calculateLib

**Висновок:**

Виконавши лабораторну роботу з розробки і використання динамічних бібліотек я вивчив прийоми розробки і використання процедур, представлених у вигляді динамічних бібліотек.  
Я на практиці ознайомився із різними варіантами підключення бібліотек, створивши 4 різні програми:

- із динамічним підключенням бібліотеки з точкою входу

- із статичним підключенням бібліотеки з точкою входу

- із динамічним підключенням бібліотеки без точки входу

- із статичним підключенням бібліотеки без точки входу.

На мою думку, метод із динамічним підключенням є кращим за статичний, оскільки в ньому не потрібен окремий файл .def, який ускладнює внесення змін до програми.

Отже, такий підхід декомпозиції коду зі створенням бібліотек є з одної сторони складнішим у вивченні та реалізації, ніж створення окремих процедур (як у 7 лабораторній), але більш гнучким та зручним у використанні.