**Міністерство освіти і науки України**  
 **Національний технічний університет України**  
 **«Київський політехнічний інститут імені Ігоря Сікорського»**  
 **Факультет інформатики та обчислювальної техніки**  
 **Кафедра обчислювальної техніки**

**Лабораторна робота №8**

з дисципліни  
 «Системне програмування»

на тему

“Розробка та використання динамічних бібліотек”

Виконав: Перевірив:

Студент 2-го курсу групи ІМ-12 Павлов Валерій Георгійович

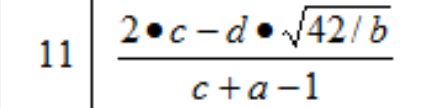
Креславський Михайло Олегович  
номер у списку групи: 11

**Київ 2023**

**Мета роботи**

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

Формула:



**Лістинги програм**

**Неявне без точки входу**

Основний файл

.386

.model flat, stdcall

option casemap: none

include \masm32\include\masm32rt.inc

; Functions: MessageBox, FloatToStr2, wsprintf, LoadLibrary, GetProcAddress, szCatStr, FreeLibrary, ExitProcess

displayMessage macro windowMessage, windowTitle

invoke MessageBox, 0, offset windowMessage, offset windowTitle, 0

endm

.data

labTitle db "System programming 8 lab", 0

labMessageFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = %s", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10, 0

errorZeroDenominatorFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"DENOMINATOR IS 0", 10,

"DIVISION BY 0 IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

errorAreaOfDefinitionFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"TAKING SQUARE ROOT OF NEGATIVE NUMBER IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

procedure db "calculate", 0

dll db "8-11-IM-12-Kreslavskyi-dll", 0

arrayA dq 2.8, 0.4, 1.1, 0.2, 2.4, -0.2

arrayB dq 1.6, 1.3, 2.9, -2.4, 3.3, 14.7

arrayC dq 0.6, 0.6, -0.5, -0.3, 1.8, -5.4

arrayD dq 1.9, 0.9, -2.1, -2.9, -4.9, 6.2

one dq 1.0

two dq 2.0

fourtyTwo dq 42.0

zero dq 0.0

.data?

result dq 1 dup(?)

numerator dt 16 dup(?)

denominator dt 16 dup(?)

final db 64 dup(?)

numberA db 32 dup(?)

numberB db 32 dup(?)

numberC db 32 dup(?)

numberD db 32 dup(?)

labMessage db 128 dup(?)

labMessageBuffer db 128 dup(?)

hLib dd ?

address dd ?

.code

programSixthLab:

mov ebp, 0

.while ebp < 6

invoke LoadLibrary, addr dll

mov hLib, eax

invoke GetProcAddress, hLib, addr procedure

mov address, eax

finit

invoke FloatToStr2, arrayA[ebp \* 8], addr numberA

invoke FloatToStr2, arrayB[ebp \* 8], addr numberB

invoke FloatToStr2, arrayC[ebp \* 8], addr numberC

invoke FloatToStr2, arrayD[ebp \* 8], addr numberD

fld arrayC[ebp \* 8] ;; c

fld arrayA[ebp \* 8] ;; a

fadd st(0), st(1) ;; c + a

fld one ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

ftst

fstsw ax

sahf

jz zeroDenominatorError

push offset result

push offset denominator

push offset numerator

push offset two

push offset fourtyTwo

push offset one

lea ecx, arrayD[ebp\*8]

push ecx

lea ecx, arrayC[ebp\*8]

push ecx

lea ecx, arrayB[ebp\*8]

push ecx

lea ecx, arrayA[ebp\*8]

push ecx

call [address]

fld fourtyTwo ;; 42

fdiv arrayB[ebp \* 8] ;; 42/b

fsqrt ;; sqrt(42/b)

ftst

fnstsw ax

sahf

test ah, 01000000b

jnz errorAreaOfDefinitionError

invoke FloatToStr2, result, addr final

invoke wsprintf, addr labMessageBuffer, addr labMessageFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr final,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

zeroDenominatorError:

invoke wsprintf, addr labMessageBuffer, addr errorZeroDenominatorFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

errorAreaOfDefinitionError:

invoke wsprintf, addr labMessageBuffer, addr errorAreaOfDefinitionFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

nextTick:

mov labMessage, 0h

inc ebp

.endw

invoke FreeLibrary, hLib

invoke ExitProcess, 0

end programSixthLab

Файл з процедурою

.386

.model flat, stdcall

option casemap :none

.code

calculate proc arrayA: ptr qword, arrayB: ptr qword, arrayC: ptr qword, arrayD: ptr qword,

one: ptr qword, fourtyTwo: ptr qword, two: ptr qword, numerator: ptr tbyte, denominator: ptr tbyte, result: ptr qword

finit

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, arrayA

fld qword ptr [ecx] ;; a

fadd st(0), st(1) ;; c + a

mov ecx, one

fld qword ptr [ecx] ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

mov ecx, denominator

fstp tbyte ptr [ecx] ;; save denominator

mov ecx, fourtyTwo

fld qword ptr [ecx] ;; 42

mov ecx, arrayB

fdiv qword ptr [ecx] ;; 42/b

fsqrt ;; sqrt(42/b)

mov ecx, arrayD

fmul qword ptr [ecx] ;; d\*sqrt(42/b)

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, two

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

fsub st(0), st(1) ;; 2\*c - d\*sqrt(42/b)

mov ecx, numerator

fstp tbyte ptr [ecx] ;; save numerator

mov ecx, numerator

fld tbyte ptr [ecx] ; st(1)

mov ecx, denominator

fld tbyte ptr [ecx] ; st(0)

fdivp st(1), st(0)

mov ecx, result

fstp qword ptr [ecx]

ret 40

calculate endp

end

BAT файл

\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi-dll.asm"

\masm32\bin\Link.exe /out:"8-11-IM-12-Kreslavskyi-dll.dll" /export:calculate /dll /noentry "8-11-IM-12-Kreslavskyi-dll.obj"

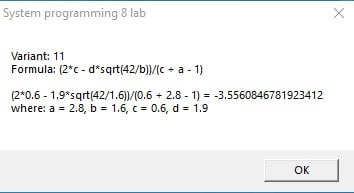
\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi.asm"

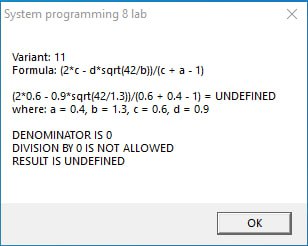
\masm32\bin\Link.exe /subsystem:windows "8-11-IM-12-Kreslavskyi.obj"

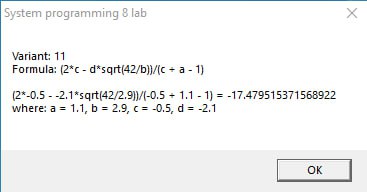
8-11-IM-12-Kreslavskyi.exe

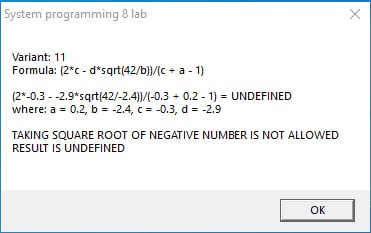
**Скріншоти виконання програми та контрольні розрахунки**

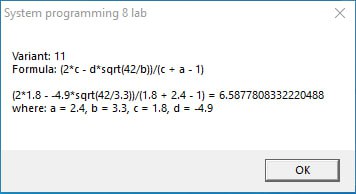
(зеленим підкреслено ті цифри, що співпадають, червоним – ті, що ні)

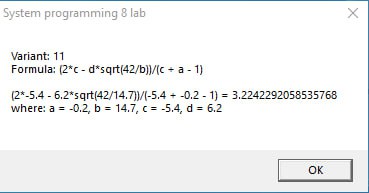
****

****

****

****

****



**Неявне з точкою входу**

Основний файл

.386

.model flat, stdcall

option casemap: none

include \masm32\include\masm32rt.inc

; Functions: MessageBox, FloatToStr2, wsprintf, LoadLibrary, GetProcAddress, szCatStr, FreeLibrary, ExitProcess

displayMessage macro windowMessage, windowTitle

invoke MessageBox, 0, offset windowMessage, offset windowTitle, 0

endm

.data

labTitle db "System programming 8 lab", 0

labMessageFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = %s", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10, 0

errorZeroDenominatorFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"DENOMINATOR IS 0", 10,

"DIVISION BY 0 IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

errorAreaOfDefinitionFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"TAKING SQUARE ROOT OF NEGATIVE NUMBER IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

procedure db "calculate", 0

dll db "8-11-IM-12-Kreslavskyi-dll", 0

arrayA dq 2.8, 0.4, 1.1, 0.2, 2.4, -0.2

arrayB dq 1.6, 1.3, 2.9, -2.4, 3.3, 14.7

arrayC dq 0.6, 0.6, -0.5, -0.3, 1.8, -5.4

arrayD dq 1.9, 0.9, -2.1, -2.9, -4.9, 6.2

one dq 1.0

two dq 2.0

fourtyTwo dq 42.0

zero dq 0.0

.data?

result dq 1 dup(?)

numerator dt 16 dup(?)

denominator dt 16 dup(?)

final db 64 dup(?)

numberA db 32 dup(?)

numberB db 32 dup(?)

numberC db 32 dup(?)

numberD db 32 dup(?)

labMessage db 128 dup(?)

labMessageBuffer db 128 dup(?)

hLib dd ?

address dd ?

.code

programSixthLab:

mov ebp, 0

.while ebp < 6

invoke LoadLibrary, addr dll

mov hLib, eax

invoke GetProcAddress, hLib, addr procedure

mov address, eax

finit

invoke FloatToStr2, arrayA[ebp \* 8], addr numberA

invoke FloatToStr2, arrayB[ebp \* 8], addr numberB

invoke FloatToStr2, arrayC[ebp \* 8], addr numberC

invoke FloatToStr2, arrayD[ebp \* 8], addr numberD

fld arrayC[ebp \* 8] ;; c

fld arrayA[ebp \* 8] ;; a

fadd st(0), st(1) ;; c + a

fld one ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

ftst

fstsw ax

sahf

jz zeroDenominatorError

push offset result

push offset denominator

push offset numerator

push offset two

push offset fourtyTwo

push offset one

lea ecx, arrayD[ebp\*8]

push ecx

lea ecx, arrayC[ebp\*8]

push ecx

lea ecx, arrayB[ebp\*8]

push ecx

lea ecx, arrayA[ebp\*8]

push ecx

call [address]

fld fourtyTwo ;; 42

fdiv arrayB[ebp \* 8] ;; 42/b

fsqrt ;; sqrt(42/b)

ftst

fnstsw ax

sahf

test ah, 01000000b

jnz errorAreaOfDefinitionError

invoke FloatToStr2, result, addr final

invoke wsprintf, addr labMessageBuffer, addr labMessageFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr final,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

zeroDenominatorError:

invoke wsprintf, addr labMessageBuffer, addr errorZeroDenominatorFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

errorAreaOfDefinitionError:

invoke wsprintf, addr labMessageBuffer, addr errorAreaOfDefinitionFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

nextTick:

mov labMessage, 0h

inc ebp

.endw

invoke FreeLibrary, hLib

invoke ExitProcess, 0

end programSixthLab

Файл з процедурою

.386

.model flat, stdcall

option casemap :none

.code

entryPoint proc hInstDLL: dword, reason:dword

mov eax, 1

ret

entryPoint endp

calculate proc arrayA: ptr qword, arrayB: ptr qword, arrayC: ptr qword, arrayD: ptr qword,

one: ptr qword, fourtyTwo: ptr qword, two: ptr qword, numerator: ptr tbyte, denominator: ptr tbyte, result: ptr qword

finit

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, arrayA

fld qword ptr [ecx] ;; a

fadd st(0), st(1) ;; c + a

mov ecx, one

fld qword ptr [ecx] ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

mov ecx, denominator

fstp tbyte ptr [ecx] ;; save denominator

mov ecx, fourtyTwo

fld qword ptr [ecx] ;; 42

mov ecx, arrayB

fdiv qword ptr [ecx] ;; 42/b

fsqrt ;; sqrt(42/b)

mov ecx, arrayD

fmul qword ptr [ecx] ;; d\*sqrt(42/b)

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, two

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

fsub st(0), st(1) ;; 2\*c - d\*sqrt(42/b)

mov ecx, numerator

fstp tbyte ptr [ecx] ;; save numerator

mov ecx, numerator

fld tbyte ptr [ecx] ; st(1)

mov ecx, denominator

fld tbyte ptr [ecx] ; st(0)

fdivp st(1), st(0)

mov ecx, result

fstp qword ptr [ecx]

ret 40

calculate endp

end entryPoint

BAT файл

\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi-dll.asm"

\masm32\bin\Link.exe /out:"8-11-IM-12-Kreslavskyi-dll.dll" /export:calculate /dll "8-11-IM-12-Kreslavskyi-dll.obj"

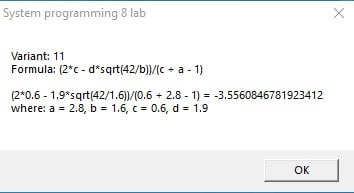
\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi.asm"

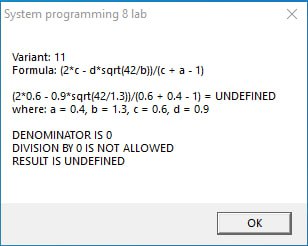
\masm32\bin\Link.exe /subsystem:windows "8-11-IM-12-Kreslavskyi.obj"

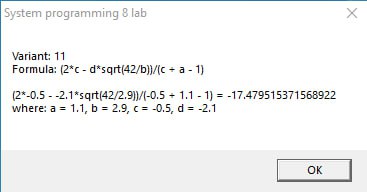
8-11-IM-12-Kreslavskyi.exe

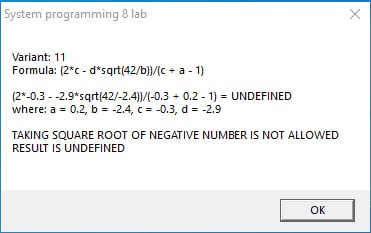
**Скріншоти виконання програми та контрольні розрахунки**

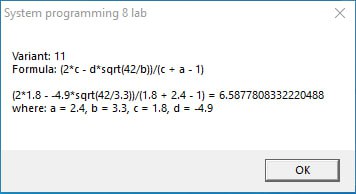
(зеленим підкреслено ті цифри, що співпадають, червоним – ті, що ні)

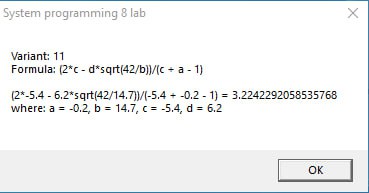
****

****

****

****

****

****

**Явне без точки входу**

Основний файл

.386

.model flat, stdcall

option casemap: none

include \masm32\include\masm32rt.inc

; Functions: MessageBox, FloatToStr2, wsprintf, szCatStr, ExitProcess

includelib 8-11-IM-12-Kreslavskyi-dll.lib

calculate proto :ptr qword, :ptr qword, :ptr qword, :ptr qword,

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

:ptr tbyte, :ptr tbyte, :ptr qword

displayMessage macro windowMessage, windowTitle

invoke MessageBox, 0, offset windowMessage, offset windowTitle, 0

endm

.data

labTitle db "System programming 8 lab", 0

labMessageFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = %s", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10, 0

errorZeroDenominatorFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"DENOMINATOR IS 0", 10,

"DIVISION BY 0 IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

errorAreaOfDefinitionFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"TAKING SQUARE ROOT OF NEGATIVE NUMBER IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

arrayA dq 2.8, 0.4, 1.1, 0.2, 2.4, -0.2

arrayB dq 1.6, 1.3, 2.9, -2.4, 3.3, 14.7

arrayC dq 0.6, 0.6, -0.5, -0.3, 1.8, -5.4

arrayD dq 1.9, 0.9, -2.1, -2.9, -4.9, 6.2

one dq 1.0

two dq 2.0

fourtyTwo dq 42.0

zero dq 0.0

.data?

result dq 1 dup(?)

numerator dt 16 dup(?)

denominator dt 16 dup(?)

final db 64 dup(?)

numberA db 32 dup(?)

numberB db 32 dup(?)

numberC db 32 dup(?)

numberD db 32 dup(?)

labMessage db 128 dup(?)

labMessageBuffer db 128 dup(?)

.code

programSixthLab:

mov ebp, 0

.while ebp < 6

invoke FloatToStr2, arrayA[ebp \* 8], addr numberA

invoke FloatToStr2, arrayB[ebp \* 8], addr numberB

invoke FloatToStr2, arrayC[ebp \* 8], addr numberC

invoke FloatToStr2, arrayD[ebp \* 8], addr numberD

fld arrayC[ebp \* 8] ;; c

fld arrayA[ebp \* 8] ;; a

fadd st(0), st(1) ;; c + a

fld one ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

ftst

fstsw ax

sahf

jz zeroDenominatorError

invoke calculate, addr arrayA[ebp\*8], addr arrayB[ebp\*8], addr arrayC[ebp\*8], addr arrayD[ebp\*8],

addr one, addr fourtyTwo, addr two, addr numerator, addr denominator, addr result

fld fourtyTwo ;; 42

fdiv arrayB[ebp \* 8] ;; 42/b

fsqrt ;; sqrt(42/b)

ftst

fnstsw ax

sahf

test ah, 01000000b

jnz errorAreaOfDefinitionError

invoke FloatToStr2, result, addr final

invoke wsprintf, addr labMessageBuffer, addr labMessageFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr final,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

zeroDenominatorError:

invoke wsprintf, addr labMessageBuffer, addr errorZeroDenominatorFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

errorAreaOfDefinitionError:

invoke wsprintf, addr labMessageBuffer, addr errorAreaOfDefinitionFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

nextTick:

mov labMessage, 0h

inc ebp

.endw

invoke ExitProcess, 0

end programSixthLab

Файл з процедурою

.386

.model flat, stdcall

option casemap :none

.code

calculate proc arrayA: ptr qword, arrayB: ptr qword, arrayC: ptr qword, arrayD: ptr qword,

one: ptr qword, fourtyTwo: ptr qword, two: ptr qword, numerator: ptr tbyte, denominator: ptr tbyte, result: ptr qword

finit

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, arrayA

fld qword ptr [ecx] ;; a

fadd st(0), st(1) ;; c + a

mov ecx, one

fld qword ptr [ecx] ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

mov ecx, denominator

fstp tbyte ptr [ecx] ;; save denominator

mov ecx, fourtyTwo

fld qword ptr [ecx] ;; 42

mov ecx, arrayB

fdiv qword ptr [ecx] ;; 42/b

fsqrt ;; sqrt(42/b)

mov ecx, arrayD

fmul qword ptr [ecx] ;; d\*sqrt(42/b)

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, two

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

fsub st(0), st(1) ;; 2\*c - d\*sqrt(42/b)

mov ecx, numerator

fstp tbyte ptr [ecx] ;; save numerator

mov ecx, numerator

fld tbyte ptr [ecx] ; st(1)

mov ecx, denominator

fld tbyte ptr [ecx] ; st(0)

fdivp st(1), st(0)

mov ecx, result

fstp qword ptr [ecx]

ret

calculate endp

end

BAT файл

\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi-dll.asm"

\masm32\bin\Link.exe /out:"8-11-IM-12-Kreslavskyi-dll.dll" /def:"8-11-IM-12-Kreslavskyi.def" /dll /noentry "8-11-IM-12-Kreslavskyi-dll.obj"

\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi.asm"

\masm32\bin\Link.exe /subsystem:windows "8-11-IM-12-Kreslavskyi.obj"

8-11-IM-12-Kreslavskyi.exe

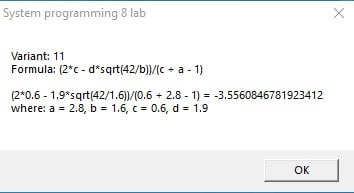
DEF файл

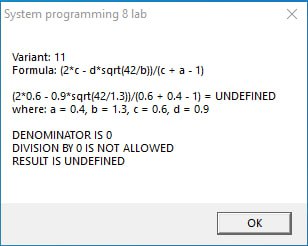
LIBRARY 8-11-IM-12-Kreslavskyi-dll

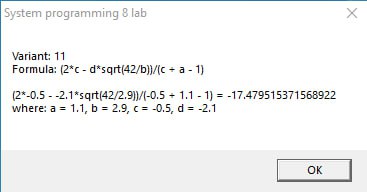
EXPORTS calculate

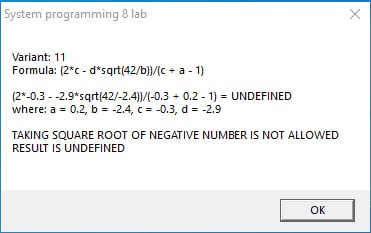
**Скріншоти виконання програми та контрольні розрахунки**

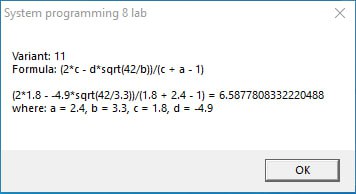
(зеленим підкреслено ті цифри, що співпадають, червоним – ті, що ні)

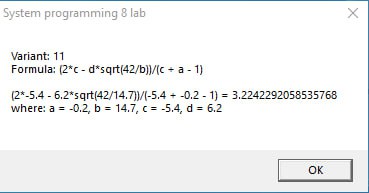
****

****

****

****

****

****

**Явне з точкою входу**

Основний файл

.386

.model flat, stdcall

option casemap: none

include \masm32\include\masm32rt.inc

; Functions: MessageBox, FloatToStr2, wsprintf, szCatStr, ExitProcess

includelib 8-11-IM-12-Kreslavskyi-dll.lib

calculate proto :ptr qword, :ptr qword, :ptr qword, :ptr qword,

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

:ptr tbyte, :ptr tbyte, :ptr qword

displayMessage macro windowMessage, windowTitle

invoke MessageBox, 0, offset windowMessage, offset windowTitle, 0

endm

.data

labTitle db "System programming 8 lab", 0

labMessageFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = %s", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10, 0

errorZeroDenominatorFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"DENOMINATOR IS 0", 10,

"DIVISION BY 0 IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

errorAreaOfDefinitionFormat db "Variant: 11", 10,

"Formula: (2\*c - d\*sqrt(42/b))/(c + a - 1)", 10, 10,

"(2\*%s - %s\*sqrt(42/%s))/(%s + %s - 1) = UNDEFINED", 10,

"where: a = %s, b = %s, c = %s, d = %s", 10, 10,

"TAKING SQUARE ROOT OF NEGATIVE NUMBER IS NOT ALLOWED", 10,

"RESULT IS UNDEFINED", 10, 10, 0

arrayA dq 2.8, 0.4, 1.1, 0.2, 2.4, -0.2

arrayB dq 1.6, 1.3, 2.9, -2.4, 3.3, 14.7

arrayC dq 0.6, 0.6, -0.5, -0.3, 1.8, -5.4

arrayD dq 1.9, 0.9, -2.1, -2.9, -4.9, 6.2

one dq 1.0

two dq 2.0

fourtyTwo dq 42.0

zero dq 0.0

.data?

result dq 1 dup(?)

numerator dt 16 dup(?)

denominator dt 16 dup(?)

final db 64 dup(?)

numberA db 32 dup(?)

numberB db 32 dup(?)

numberC db 32 dup(?)

numberD db 32 dup(?)

labMessage db 128 dup(?)

labMessageBuffer db 128 dup(?)

.code

programSixthLab:

mov ebp, 0

.while ebp < 6

invoke FloatToStr2, arrayA[ebp \* 8], addr numberA

invoke FloatToStr2, arrayB[ebp \* 8], addr numberB

invoke FloatToStr2, arrayC[ebp \* 8], addr numberC

invoke FloatToStr2, arrayD[ebp \* 8], addr numberD

fld arrayC[ebp \* 8] ;; c

fld arrayA[ebp \* 8] ;; a

fadd st(0), st(1) ;; c + a

fld one ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

ftst

fstsw ax

sahf

jz zeroDenominatorError

invoke calculate, addr arrayA[ebp\*8], addr arrayB[ebp\*8], addr arrayC[ebp\*8], addr arrayD[ebp\*8],

addr one, addr fourtyTwo, addr two, addr numerator, addr denominator, addr result

fld fourtyTwo ;; 42

fdiv arrayB[ebp \* 8] ;; 42/b

fsqrt ;; sqrt(42/b)

ftst

fnstsw ax

sahf

test ah, 01000000b

jnz errorAreaOfDefinitionError

invoke FloatToStr2, result, addr final

invoke wsprintf, addr labMessageBuffer, addr labMessageFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr final,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

zeroDenominatorError:

invoke wsprintf, addr labMessageBuffer, addr errorZeroDenominatorFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

errorAreaOfDefinitionError:

invoke wsprintf, addr labMessageBuffer, addr errorAreaOfDefinitionFormat,

addr numberC, addr numberD, addr numberB,

addr numberC, addr numberA,

addr numberA, addr numberB, addr numberC, addr numberD

invoke szCatStr, addr labMessage, addr labMessageBuffer

displayMessage labMessage, labTitle

jmp nextTick

nextTick:

mov labMessage, 0h

inc ebp

.endw

invoke ExitProcess, 0

end programSixthLab

Файл з процедурою

.386

.model flat, stdcall

option casemap :none

.code

entryPoint proc hInstDLL: dword, reason:dword

mov eax, 1

ret

entryPoint endp

calculate proc arrayA: ptr qword, arrayB: ptr qword, arrayC: ptr qword, arrayD: ptr qword,

one: ptr qword, fourtyTwo: ptr qword, two: ptr qword, numerator: ptr tbyte, denominator: ptr tbyte, result: ptr qword

finit

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, arrayA

fld qword ptr [ecx] ;; a

fadd st(0), st(1) ;; c + a

mov ecx, one

fld qword ptr [ecx] ;; 1

fxch st(1)

fsub st(0), st(1) ;; c + a - 1

mov ecx, denominator

fstp tbyte ptr [ecx] ;; save denominator

mov ecx, fourtyTwo

fld qword ptr [ecx] ;; 42

mov ecx, arrayB

fdiv qword ptr [ecx] ;; 42/b

fsqrt ;; sqrt(42/b)

mov ecx, arrayD

fmul qword ptr [ecx] ;; d\*sqrt(42/b)

mov ecx, arrayC

fld qword ptr [ecx] ;; c

mov ecx, two

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

fsub st(0), st(1) ;; 2\*c - d\*sqrt(42/b)

mov ecx, numerator

fstp tbyte ptr [ecx] ;; save numerator

mov ecx, numerator

fld tbyte ptr [ecx] ; st(1)

mov ecx, denominator

fld tbyte ptr [ecx] ; st(0)

fdivp st(1), st(0)

mov ecx, result

fstp qword ptr [ecx]

ret

calculate endp

end entryPoint

BAT файл

\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi-dll.asm"

\masm32\bin\Link.exe /out:"8-11-IM-12-Kreslavskyi-dll.dll" /dll /def:"8-11-IM-12-Kreslavskyi.def" "8-11-IM-12-Kreslavskyi-dll.obj"

\masm32\bin\ml /c /coff "8-11-IM-12-Kreslavskyi.asm"

\masm32\bin\Link.exe /subsystem:windows "8-11-IM-12-Kreslavskyi.obj"

8-11-IM-12-Kreslavskyi.exe

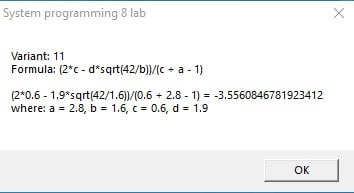
DEF файл

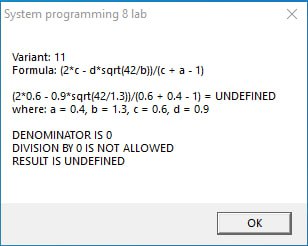
LIBRARY 8-11-IM-12-Kreslavskyi-dll

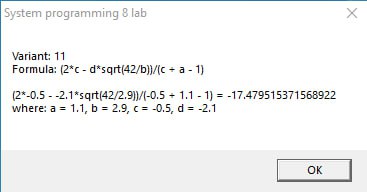
EXPORTS calculate

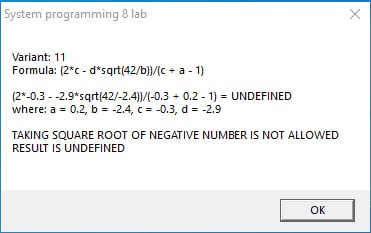
**Скріншоти виконання програми та контрольні розрахунки**

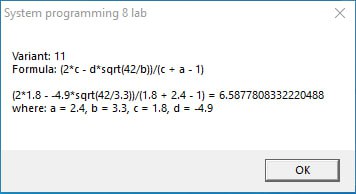
(зеленим підкреслено ті цифри, що співпадають, червоним – ті, що ні)

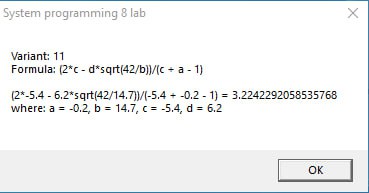
****

****

****

****

****



**Висновок**

Під час виконання лабораторної роботи я дослідив прийоми розробки та використання динамічних бібліотек. Основна мета полягала у тому, щоб ознайомитися з методами створення та використання DLL-бібліотек та реалізувати процедури у вигляді таких бібліотек.

Я ознайомився з методами створення та використання DLL-бібліотек, вивчив способи з та без вказівки точки входу.

Для перевірки правильності розрахунків я виконав контрольні розрахунки і порівняв їх з результатами, що виводила програма. Також проведено відладку програми шляхом порівняння розрахунків з контрольними прикладами.

У підсумку, завершивши лабораторну роботу, я зробив наступні висновки. Вивчення та використання динамічних бібліотек є важливою складовою системного програмування. Вони надають зручність та гнучкість у розробці програмного забезпечення, дозволяючи використовувати перевикористовувані процедури та функціональні блоки.

Процес розробки DLL-бібліотеки включає написання процедур, створення файлу налаштувань модуля та компіляцію. Якщо потрібно, можна використовувати явне або неявне завантаження бібліотеки в основну програму.

Ця лабораторна робота надала мені практичні навички розробки та використання динамічних бібліотек. Успішно були реалізовані процедури обчислення заданого арифметичного виразу та виведення результатів, а також була проведена перевірка правильності обчислень за допомогою контрольних розрахунків.