**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 і використання динамічних бібліотек”

Виконав: Перевірив:

Студент ІІ курсу ФІОТ доц. Павлов В. Г.

групи ІМ-12

Сутулов Нікіта Олегович

номер у списку групи (варіант): 20

номер залікової книжки: 1229

Київ 2023

**Мета роботи:** Вивчення прийомів розробки і використання процедур, представлених у вигляді динамічних бібліотек.

Завдання за варіантом:

номер у списку групи: 20, тож 20 варіант: (arctg(2\*c) / d + 2) / (b - a - 1)

**Контрольні приклади:**

1. Чисельник і знаменник більші за нуль.

a = 4.1, b = 5.4, c = 3.3, d = 1.5.

(arctg(2\*3.3) / 1.5 + 2) / (5.4 - 4.1 - 1) = (arctg(6.6) / 1.5 + 2) / 0.3 =

= (1.42042489879 / 1.5 + 2) / 0.3 = (0.946949932525 + 2) / 0.3 =

= 2.946949932525 / 0.3 = 9.82316644175

Таким чином:

Результат = 9.82316644175

2. Чисельник та знаменник менші за нуль.

a = -5.9, b = -7.4, c = 4.3, d = -0.2.

(arctg(2\*4.3) / (-0.2) + 2) / (-7.4 + 5.9 - 1) = (arctg(8.6) / (-0.2) + 2) / (-2.5) =

= (1.45503710907 / (-0.2) + 2) / (-2.5) = (-7.27518554537 + 2) / (-2.5) =

= -5.27518554537 / (-2.5) = 2.11007421815

Таким чином:

Результат = 2.11007421815

3. Знаменник дорівнює нулю.

a = 19.6, b = 20.6, c = -40.5, d = 19.2.

b - a - 1 = 20.6 - 19.6 - 1 = 1 - 1 = 0.

4. Чисельник більший за нуль, а знаменник менший за нуль.

a = 3.9, b = 4.3, c = -5.4, d = -2.5.

(arctg(2\*(-5.4)) / (-2.5) + 2) / (4.3 - 3.9 - 1) = (arctg(-10.8) / (-2.5) + 2) / (-0.6) =

= (-1.47846699206 / (-2.5) + 2) / (-0.6) = (0.591386796825 + 2) / (-0.6) =

= 2.591386796825 / (-0.6) = -4.31897799471

Таким чином:

Результат = -4.31897799471

5. Чисельник менший за нуль, а знаменник більший за нуль.

a = 2.8, b = 7.9, c = -8.8, d = 0.1.

(arctg(2\*(-8.8)) / 0.1 + 2) / (7.9 - 2.8 - 1) = (arctg(-17.6) / 0.1 + 2) / 4.1 =

= (-1.51403916897 / 0.1 + 2) / 4.1 = (-15.1403916897 + 2) / 4.1 =

= -13.1403916897 / 4.1 = -3.20497358286

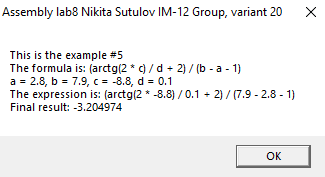
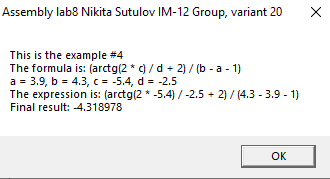
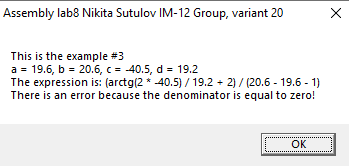
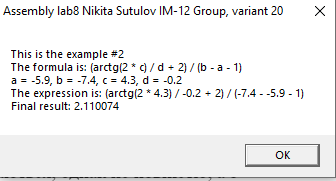
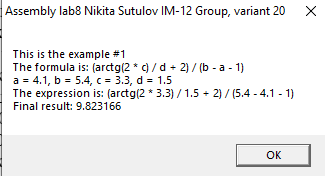
Таким чином:

Результат = -3.20497358286

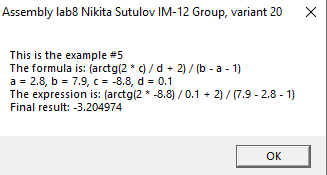
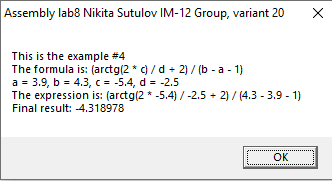
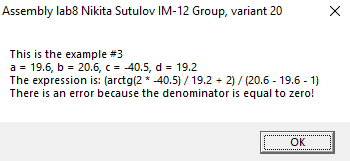
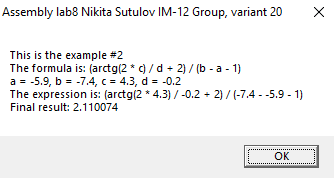
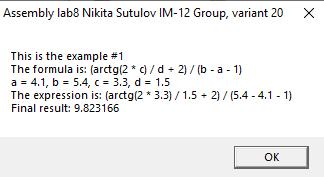
У формулі використовується функція arctg, яка визначена для всієї множини дійсних чисел, а тому додаткова перевірка відповідності аргумента функції її області визначення, а разом із нею й відповідний контрольний приклад, де цю область визначення порушено, НЕ ПОТРІБНІ.

**Демонстрація роботи програми**

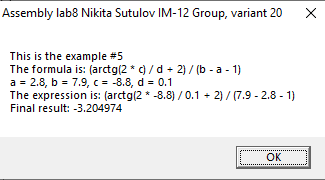
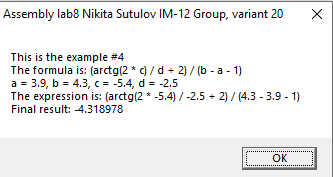
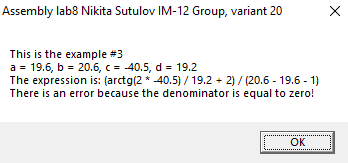
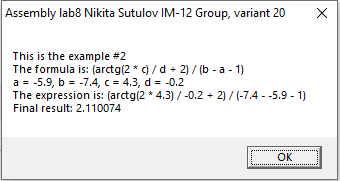
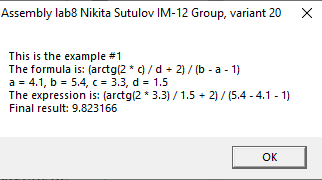
Implicit-entry:



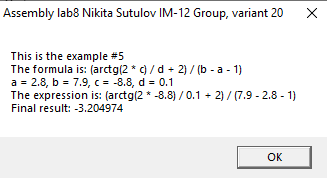
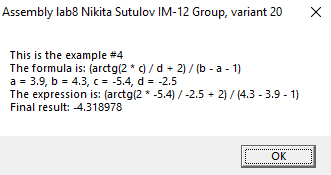
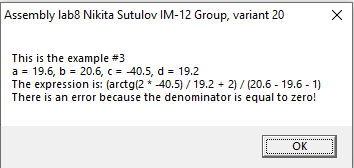
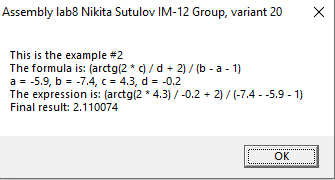
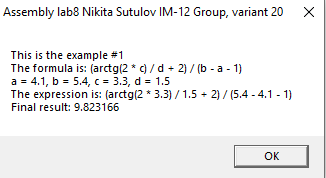
Implicit-noentry:



Explicit-entry:



Explicit-noentry:



В результаті порівняння контрольних розрахунків з результатами виконання програми видно, що дані збігаються, однак не повністю, а з деякою точністю, так як при контрольних розрахунках виконано округлення до 11 знаків після коми, а під час виконання програми комп’ютер здійснює округлення до 6 знаків після коми.

Отже програма в усіх варіантах формування та підключення динамічних бібліотек (неявне завантаження з точкою входу, неявне завантаження без точки входу, явне завантаження з точкою входу, явне завантаження без точки входу) працює коректно в усіх сценаріях:

* чисельник та знаменник більше за 0;
* чисельник та знаменник менше за 0;
* чисельник більше за 0, а знаменник – менше;
* чисельник менше за 0, а знаменник – більше;
* і також той випадок, коли знаменник дорівнює 0, опрацьовується коректно (виводиться повідомлення з помилкою). Тут варто також зауважити, що в цьому випадку не відбувається блокування розрахунків для наступних прикладів (якщо в них знаменник не дорівнює нулю), тож у них результат виконання програми коректний.

**Лістинг програми**

*Файл процедури з використанням точки входу (****entry****)* 8-20-IM-12-Sutulov-lib.asm:

.386

.model flat, stdcall

option casemap :none

.data

SutulovConstZero dq 0.0

SutulovConstOne dq 1.0

SutulovConstTwo dq 2.0

.data?

SutulovDenominator dt ?

SutulovNumerator dt ?

SutulovArctanResult dt ?

SutulovArctanArgument dt ?

.code

DllMain proc hInstDLL:dword, reason:dword, unused:dword

mov eax, 1

ret

DllMain endp

SutulovPerformCalculation proc SutulovCurrentA:ptr qword, SutulovCurrentB:ptr qword, SutulovCurrentC:ptr qword, SutulovCurrentD:ptr qword, SutulovFinalResultPointer:ptr qword

xor eax, eax ; setting eax to zero

finit

; calculating denominator: b - a - 1

mov ebx, SutulovCurrentB

fld qword ptr[ebx]

mov ebx, SutulovCurrentA

fld qword ptr[ebx]

fsub

fld SutulovConstOne

fsub

; checking denominator for zero:

fcom SutulovConstZero

fstsw ax

sahf

je wrongDenominatorValue

; saving denominator to 10-bytes buffer

fstp SutulovDenominator

; calculating arctan argument: 2 \* c

fld SutulovConstTwo

mov ebx, SutulovCurrentC

fld qword ptr[ebx]

fmul

; saving arctan argument to 10-bytes buffer

fstp SutulovArctanArgument

; calculating arctg(2 \* c)

fld SutulovArctanArgument

fld SutulovConstOne

fpatan

; saving arctan calculation result to 10-bytes buffer

fstp SutulovArctanResult

; calculating numerator: (arctg(2 \* c) / d + 2)

fld SutulovArctanResult

mov ebx, SutulovCurrentD

fld qword ptr[ebx]

fdiv ; d can't be 0 according to the task, so no zero checks needed here

fld SutulovConstTwo

fadd

; saving numerator to 10-bytes buffer

fstp SutulovNumerator

; dividing numerator by denominator

fld SutulovNumerator

fld SutulovDenominator

fdiv

jmp exitFromProcedure

wrongDenominatorValue:

mov eax, 1

jmp exitFromProcedure

exitFromProcedure:

; saving the final result to 8-bytes variable

mov ebx, SutulovFinalResultPointer

fstp qword ptr [ebx]

ret

SutulovPerformCalculation endp

end DllMain

*Файл головної програми з* ***неявним*** *завантаженням бібліотеки* 8-20-IM-12-Sutulov.asm:

.386

.model flat, stdcall

option casemap :none

include \masm32\include\windows.inc

include \masm32\include\dialogs.inc

include \masm32\include\masm32.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

includelib \masm32\lib\masm32.lib ; FloatToStr

includelib \masm32\lib\user32.lib ; MessageBox, wsprintf

includelib \masm32\lib\kernel32.lib ; ExitProcess

includelib 8-20-IM-12-Sutulov-lib.lib

SutulovPerformCalculation proto :ptr qword, :ptr qword, :ptr qword, :ptr qword, :ptr qword

.data

; arrays

SutulovAArray dq 4.1, -5.9, 19.6, 3.9, 2.8

SutulovBArray dq 5.4, -7.4, 20.6, 4.3, 7.9

SutulovCArray dq 3.3, 4.3, -40.5, -5.4, -8.8

SutulovDArray dq 1.5, -0.2, 19.2, -2.5, 0.1

; window caption

SutulovWindowCaption db "Assembly lab8 Nikita Sutulov IM-12 Group, variant 20", 0

; window text template

SutulovExampleForm db "This is the example #%d", 0

SutulovFormula db "The formula is: (arctg(2 \* c) / d + 2) / (b - a - 1)", 0

SutulovCurrentValuesForm db "a = %s, b = %s, c = %s, d = %s", 0

SutulovExpressionForm db "The expression is: (arctg(2 \* %s) / %s + 2) / (%s - %s - 1)", 0

SutulovFinalResultForm db "Final result: %s", 0

SutulovErrorMessage db "There is an error because the denominator is equal to zero!", 0

SutulovFinalForm db "%s", 10 ,13,

"%s", 10, 13,

"%s", 10, 13,

"%s", 10, 13,

"%s", 0

SutulovErrorForm db "%s", 10, 13,

"%s", 10, 13,

"%s", 10, 13,

"%s", 0

SutulovErrorMessageForm db "%s", 0

.data?

; buffers

SutulovExampleBuffer db 64 dup (?)

SutulovCurrentValuesBuffer db 64 dup (?)

SutulovExpressionBuffer db 128 dup (?)

SutulovFinalResultBuffer db 64 dup (?)

SutulovFinalBuffer db 256 dup (?)

SutulovFinalResult dq ?

SutulovFinalResultPointer dd ?

SutulovCurrentAString db 16 dup (?)

SutulovCurrentBString db 16 dup (?)

SutulovCurrentCString db 16 dup (?)

SutulovCurrentDString db 16 dup (?)

SutulovFinalResultString db 16 dup (?)

.code

main:

lea ecx, SutulovFinalResult

mov SutulovFinalResultPointer, ecx

mov edi, 0 ; counter for indexes

mov esi, 1 ; counter for example numbers

loopToIterateThroughArrays:

cmp edi, 5

je exitLoop

invoke SutulovPerformCalculation, addr SutulovAArray[edi\*8], addr SutulovBArray[edi\*8],

addr SutulovCArray[edi\*8], addr SutulovDArray[edi\*8], SutulovFinalResultPointer

cmp eax, 1

je denominatorZero

normalWindow:

; converting float numbers to strings for them to be shown correctly

invoke FloatToStr, SutulovAArray[edi \* 8], offset SutulovCurrentAString

invoke FloatToStr, SutulovBArray[edi \* 8], offset SutulovCurrentBString

invoke FloatToStr, SutulovCArray[edi \* 8], offset SutulovCurrentCString

invoke FloatToStr, SutulovDArray[edi \* 8], offset SutulovCurrentDString

invoke FloatToStr, SutulovFinalResult, offset SutulovFinalResultString

; making the window text

invoke wsprintf, offset SutulovExampleBuffer,

offset SutulovExampleForm, esi

invoke wsprintf, offset SutulovCurrentValuesBuffer,

offset SutulovCurrentValuesForm,

offset SutulovCurrentAString, offset SutulovCurrentBString,

offset SutulovCurrentCString, offset SutulovCurrentDString

invoke wsprintf, offset SutulovExpressionBuffer,

offset SutulovExpressionForm,

offset SutulovCurrentCString, offset SutulovCurrentDString,

offset SutulovCurrentBString, offset SutulovCurrentAString

invoke wsprintf, offset SutulovFinalResultBuffer,

offset SutulovFinalResultForm, offset SutulovFinalResultString

invoke wsprintf, offset SutulovFinalBuffer, offset SutulovFinalForm,

offset SutulovExampleBuffer, offset SutulovFormula,

offset SutulovCurrentValuesBuffer, offset SutulovExpressionBuffer,

offset SutulovFinalResultBuffer

invoke MessageBox, 0, offset SutulovFinalBuffer, offset SutulovWindowCaption, 0

inc edi

inc esi

jmp loopToIterateThroughArrays

denominatorZero:

; converting float numbers to strings for them to be shown correctly

invoke FloatToStr, SutulovAArray[edi \* 8], offset SutulovCurrentAString

invoke FloatToStr, SutulovBArray[edi \* 8], offset SutulovCurrentBString

invoke FloatToStr, SutulovCArray[edi \* 8], offset SutulovCurrentCString

invoke FloatToStr, SutulovDArray[edi \* 8], offset SutulovCurrentDString

invoke wsprintf, offset SutulovExampleBuffer,

offset SutulovExampleForm, esi

invoke wsprintf, offset SutulovCurrentValuesBuffer,

offset SutulovCurrentValuesForm,

offset SutulovCurrentAString, offset SutulovCurrentBString,

offset SutulovCurrentCString, offset SutulovCurrentDString

invoke wsprintf, offset SutulovExpressionBuffer,

offset SutulovExpressionForm,

offset SutulovCurrentCString, offset SutulovCurrentDString,

offset SutulovCurrentBString, offset SutulovCurrentAString

invoke wsprintf, offset SutulovFinalResultBuffer,

offset SutulovErrorMessageForm, offset SutulovErrorMessage

invoke wsprintf, offset SutulovFinalBuffer, offset SutulovErrorForm,

offset SutulovExampleBuffer,

offset SutulovCurrentValuesBuffer,

offset SutulovExpressionBuffer,

offset SutulovFinalResultBuffer

invoke MessageBox, 0, offset SutulovFinalBuffer, offset SutulovWindowCaption, 0

inc edi

inc esi

jmp loopToIterateThroughArrays

exitLoop:

invoke ExitProcess, 0

end main

*Файл процедури без використання точки входу (****noentry****)* 8-20-IM-12-Sutulov-lib.asm:

.386

.model flat, stdcall

option casemap :none

.data

SutulovConstZero dq 0.0

SutulovConstOne dq 1.0

SutulovConstTwo dq 2.0

.data?

SutulovDenominator dt ?

SutulovNumerator dt ?

SutulovArctanResult dt ?

SutulovArctanArgument dt ?

.code

SutulovPerformCalculation proc SutulovCurrentA:ptr qword, SutulovCurrentB:ptr qword, SutulovCurrentC:ptr qword, SutulovCurrentD:ptr qword, SutulovFinalResultPointer:ptr qword

xor eax, eax ; setting eax to zero

finit

; calculating denominator: b - a - 1

mov ebx, SutulovCurrentB

fld qword ptr[ebx]

mov ebx, SutulovCurrentA

fld qword ptr[ebx]

fsub

fld SutulovConstOne

fsub

; checking denominator for zero:

fcom SutulovConstZero

fstsw ax

sahf

je wrongDenominatorValue

; saving denominator to 10-bytes buffer

fstp SutulovDenominator

; calculating arctan argument: 2 \* c

fld SutulovConstTwo

mov ebx, SutulovCurrentC

fld qword ptr[ebx]

fmul

; saving arctan argument to 10-bytes buffer

fstp SutulovArctanArgument

; calculating arctg(2 \* c)

fld SutulovArctanArgument

fld SutulovConstOne

fpatan

; saving arctan calculation result to 10-bytes buffer

fstp SutulovArctanResult

; calculating numerator: (arctg(2 \* c) / d + 2)

fld SutulovArctanResult

mov ebx, SutulovCurrentD

fld qword ptr[ebx]

fdiv ; d can't be 0 according to the task, so no zero checks needed here

fld SutulovConstTwo

fadd

; saving numerator to 10-bytes buffer

fstp SutulovNumerator

; dividing numerator by denominator

fld SutulovNumerator

fld SutulovDenominator

fdiv

jmp exitFromProcedure

wrongDenominatorValue:

mov eax, 1

jmp exitFromProcedure

exitFromProcedure:

; saving the final result to 8-bytes variable

mov ebx, SutulovFinalResultPointer

fstp qword ptr [ebx]

ret

SutulovPerformCalculation endp

end

*Файл головної програми з* ***явним*** *завантаженням бібліотеки* 8-20-IM-12-Sutulov.asm:

.386

.model flat, stdcall

option casemap :none

include \masm32\include\windows.inc

include \masm32\include\dialogs.inc

include \masm32\include\masm32.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

includelib \masm32\lib\masm32.lib ; FloatToStr

includelib \masm32\lib\user32.lib ; MessageBox, wsprintf

includelib \masm32\lib\kernel32.lib ; ExitProcess, LoadLibrary, GetProcAddress, FreeLibrary

includelib 8-20-IM-12-Sutulov-lib.lib

.data

; arrays

SutulovAArray dq 4.1, -5.9, 19.6, 3.9, 2.8

SutulovBArray dq 5.4, -7.4, 20.6, 4.3, 7.9

SutulovCArray dq 3.3, 4.3, -40.5, -5.4, -8.8

SutulovDArray dq 1.5, -0.2, 19.2, -2.5, 0.1

; window caption

SutulovWindowCaption db "Assembly lab8 Nikita Sutulov IM-12 Group, variant 20", 0

; window text template

SutulovExampleForm db "This is the example #%d", 0

SutulovFormula db "The formula is: (arctg(2 \* c) / d + 2) / (b - a - 1)", 0

SutulovCurrentValuesForm db "a = %s, b = %s, c = %s, d = %s", 0

SutulovExpressionForm db "The expression is: (arctg(2 \* %s) / %s + 2) / (%s - %s - 1)", 0

SutulovFinalResultForm db "Final result: %s", 0

SutulovErrorMessage db "There is an error because the denominator is equal to zero!", 0

SutulovFinalForm db "%s", 10 ,13,

"%s", 10, 13,

"%s", 10, 13,

"%s", 10, 13,

"%s", 0

SutulovErrorForm db "%s", 10, 13,

"%s", 10, 13,

"%s", 10, 13,

"%s", 0

SutulovErrorMessageForm db "%s", 0

SutulovProcedureName db "SutulovPerformCalculation", 0

SutulovLibraryName db "8-20-IM-12-Sutulov-lib.dll", 0

.data?

; buffers

SutulovExampleBuffer db 64 dup (?)

SutulovCurrentValuesBuffer db 64 dup (?)

SutulovExpressionBuffer db 128 dup (?)

SutulovFinalResultBuffer db 64 dup (?)

SutulovFinalBuffer db 256 dup (?)

SutulovFinalResult dq ?

SutulovFinalResultPointer dd ?

SutulovCurrentAString db 16 dup (?)

SutulovCurrentBString db 16 dup (?)

SutulovCurrentCString db 16 dup (?)

SutulovCurrentDString db 16 dup (?)

SutulovFinalResultString db 16 dup (?)

SutulovPerformCalculation dd ?

SutulovLibrary dd ?

.code

main:

invoke LoadLibrary, offset SutulovLibraryName

mov SutulovLibrary, eax

invoke GetProcAddress, SutulovLibrary, offset SutulovProcedureName

mov SutulovPerformCalculation, eax

lea ecx, SutulovFinalResult

mov SutulovFinalResultPointer, ecx

mov edi, 0 ; counter for indexes

mov esi, 1 ; counter for example numbers

loopToIterateThroughArrays:

cmp edi, 5

je exitLoop

mov ebx, SutulovFinalResultPointer

push ebx

lea ebx, SutulovDArray[edi\*8]

push ebx

lea ebx, SutulovCArray[edi\*8]

push ebx

lea ebx, SutulovBArray[edi\*8]

push ebx

lea ebx, SutulovAArray[edi\*8]

push ebx

call [SutulovPerformCalculation]

cmp eax, 1

je denominatorZero

normalWindow:

; converting float numbers to strings for them to be shown correctly

invoke FloatToStr, SutulovAArray[edi \* 8], offset SutulovCurrentAString

invoke FloatToStr, SutulovBArray[edi \* 8], offset SutulovCurrentBString

invoke FloatToStr, SutulovCArray[edi \* 8], offset SutulovCurrentCString

invoke FloatToStr, SutulovDArray[edi \* 8], offset SutulovCurrentDString

invoke FloatToStr, SutulovFinalResult, offset SutulovFinalResultString

; making the window text

invoke wsprintf, offset SutulovExampleBuffer,

offset SutulovExampleForm, esi

invoke wsprintf, offset SutulovCurrentValuesBuffer,

offset SutulovCurrentValuesForm,

offset SutulovCurrentAString, offset SutulovCurrentBString,

offset SutulovCurrentCString, offset SutulovCurrentDString

invoke wsprintf, offset SutulovExpressionBuffer,

offset SutulovExpressionForm,

offset SutulovCurrentCString, offset SutulovCurrentDString,

offset SutulovCurrentBString, offset SutulovCurrentAString

invoke wsprintf, offset SutulovFinalResultBuffer,

offset SutulovFinalResultForm, offset SutulovFinalResultString

invoke wsprintf, offset SutulovFinalBuffer, offset SutulovFinalForm,

offset SutulovExampleBuffer, offset SutulovFormula,

offset SutulovCurrentValuesBuffer, offset SutulovExpressionBuffer,

offset SutulovFinalResultBuffer

invoke MessageBox, 0, offset SutulovFinalBuffer, offset SutulovWindowCaption, 0

inc edi

inc esi

jmp loopToIterateThroughArrays

denominatorZero:

; converting float numbers to strings for them to be shown correctly

invoke FloatToStr, SutulovAArray[edi \* 8], offset SutulovCurrentAString

invoke FloatToStr, SutulovBArray[edi \* 8], offset SutulovCurrentBString

invoke FloatToStr, SutulovCArray[edi \* 8], offset SutulovCurrentCString

invoke FloatToStr, SutulovDArray[edi \* 8], offset SutulovCurrentDString

invoke wsprintf, offset SutulovExampleBuffer,

offset SutulovExampleForm, esi

invoke wsprintf, offset SutulovCurrentValuesBuffer,

offset SutulovCurrentValuesForm,

offset SutulovCurrentAString, offset SutulovCurrentBString,

offset SutulovCurrentCString, offset SutulovCurrentDString

invoke wsprintf, offset SutulovExpressionBuffer,

offset SutulovExpressionForm,

offset SutulovCurrentCString, offset SutulovCurrentDString,

offset SutulovCurrentBString, offset SutulovCurrentAString

invoke wsprintf, offset SutulovFinalResultBuffer,

offset SutulovErrorMessageForm, offset SutulovErrorMessage

invoke wsprintf, offset SutulovFinalBuffer, offset SutulovErrorForm,

offset SutulovExampleBuffer,

offset SutulovCurrentValuesBuffer,

offset SutulovExpressionBuffer,

offset SutulovFinalResultBuffer

invoke MessageBox, 0, offset SutulovFinalBuffer, offset SutulovWindowCaption, 0

inc edi

inc esi

jmp loopToIterateThroughArrays

exitLoop:

invoke FreeLibrary, SutulovLibrary

invoke ExitProcess, 0

end main

*DEF-файл для неявного завантаження бібліотеки* 8-20-IM-12-Sutulov-lib.def:

LIBRARY 8-20-IM-12-Sutulov-lib

EXPORTS SutulovPerformCalculation

*BAT-файл для програми з* ***явним*** *завантаженням бібліотеки* ***з точкою входу****:*

@echo off

ml /c /coff 8-20-IM-12-Sutulov-lib.asm

link /out:8-20-IM-12-Sutulov-lib.dll /export:SutulovPerformCalculation /dll 8-20-IM-12-Sutulov-lib.obj

ml /c /coff 8-20-IM-12-Sutulov.asm

link /subsystem:windows 8-20-IM-12-Sutulov.obj

pause

*BAT-файл для програми з* ***явним*** *завантаженням бібліотеки* ***без точки входу****:*

@echo off

ml /c /coff 8-20-IM-12-Sutulov-lib.asm

link /out:8-20-IM-12-Sutulov-lib.dll /export:SutulovPerformCalculation /dll 8-20-IM-12-Sutulov-lib.obj /noentry

ml /c /coff 8-20-IM-12-Sutulov.asm

link /subsystem:windows 8-20-IM-12-Sutulov.obj

pause

*BAT-файл для програми з* ***неявним*** *завантаженням бібліотеки* ***з точкою входу****:*

@echo off

ml /c /coff 8-20-IM-12-Sutulov-lib.asm

link /out:8-20-IM-12-Sutulov-lib.dll /def:8-20-IM-12-Sutulov-lib.def /dll 8-20-IM-12-Sutulov-lib.obj

ml /c /coff 8-20-IM-12-Sutulov.asm

link /subsystem:windows 8-20-IM-12-Sutulov.obj

pause

*BAT-файл для програми з* ***неявним*** *завантаженням бібліотеки* ***без точки входу****:*

@echo off

ml /c /coff 8-20-IM-12-Sutulov-lib.asm

link /out:8-20-IM-12-Sutulov-lib.dll /def:8-20-IM-12-Sutulov-lib.def /dll 8-20-IM-12-Sutulov-lib.obj /noentry

ml /c /coff 8-20-IM-12-Sutulov.asm

link /subsystem:windows 8-20-IM-12-Sutulov.obj

pause

**Висновки**

У результаті порівняння контрольних розрахунків із результатами виконання програми для всіх передбачених лабораторною роботою сценаріїв (перевірка аргумента функції на відповідність її області визначення, а також відповідний приклад, у якому цю область визначення порушено, реалізовано не було, оскільки в моєму варіанті використовується функція arctg, яка визначена для всієї множини дійсних чисел, а тому такої перевірки не потребує):

* чисельник та знаменник більше за 0;
* чисельник та знаменник менше за 0;
* чисельник більше за 0, а знаменник – менше;
* чисельник менше за 0, а знаменник – більше;
* знаменник дорівнює 0,

бачимо, що результати обчислень співпадають з деякою точністю (похибка виникає через те, що в ході виконання програми округлення здійснювалось із меншою точністю, ніж у контрольних розрахунках), і випадок, коли знаменник дорівнює 0, обробляється коректно (і в цьому випадку не блокується виконання розрахунків для наступних прикладів, якщо вони відповідають умові, що знаменник не дорівнює нулю).

У ході виконання роботи програму з лабораторної роботи №6 було перероблено так, що код, який виконує обчислення арифметичного виразу, було винесено в процедуру, що належить окремій бібліотеці, для роботи з якою було реалізовано 4 варіанти програми:

* неявне завантаження з точкою входу,
* неявне завантаження без точки входу,
* явне завантаження з точкою входу,
* явне завантаження без точки входу.

Аналізуючи результат виконання всіх варіантів програми, я можу зробити висновок, що спосіб завантаження бібліотеки при її використанні, а також використання чи невикористання в реалізації бібліотеки точки входу для процедури не впливають на результат виконання програми, а лише вимагають різних підходів для збірки. При неявному завантаженні бібліотеки треба мати файл опису, в нашому випадку з розширенням def, у якому вказано файл бібліотеки та експортовані процедури, і його треба вказати за допомогою спеціального параметра /def у виклику лінкера. При явному завантаженні цей файл не потрібен, замість нього явно виконується експорт процедури прямо у файлі головної програми за допомогою процедур LoadLibrary та GetProcedureAddress, а також при виході з програми використовується процедура FreeLibrary для вивантаження бібліотеки. Натомість при лінкуванні треба вказати назву експортованої процедури в параметрі /export. Процедура може мати точку входу (в нашому випадку це DllMain), а може й не мати (для цього треба вказати при лінкуванні параметр /noentry). Але, знову ж таки, вибір варіанта реалізації не впливатиме на результати обчислень програми.

Таким чином, роблю висновок, що програму в усіх чотирьох варіантах написано коректно.

У ході виконання цієї лабораторної роботи мною було вивчено прийоми розробки і використання процедур, представлених у вигляді динамічних бібліотек (неявне завантаження з точкою входу, неявне завантаження без точки входу, явне завантаження з точкою входу, явне завантаження без точки входу).