**Лабораторна робота №7**

з дисципліни «Системне програмування» з теми

«Модульне програмування. Використання процедур»

Виконав: Перевірив:

Студент групи ІМ-31 доц. Павлов В. Г.

Булах Богдан Валерійович

номер в списку групи: 2

Київ 2025

**Мета роботи**

Вивчення прийомів модульного програмування, методів звернення до процедур і передачі в них параметрів.

**Порядок виконання роботи:**

1. Вивчити методи звернення до процедур і передачі в них параметрів.

2. Для свого варіанту індивідуального завдання до лабораторної роботи 6 розробити програму на мові Асемблер, в якій використовувати три процедури з різними способами передачі параметрів:

* через регістри
* через стек
* за допомогою директив EXTRN та PUBLIC

3. Для цього чисельник дробу зі свого варіанту індивідуального завдання до лабораторної роботи 6 розділити на два доданка, з яких для першого застосувати передачу параметрів і результату через регістри, а для другого – через стек. Для знаменника використовувати метод оголошення загальних змінних директивами public і extern. Виведення результату виконати в основній програмі.

4. Розрахунки (п. 3) повторити в програмі для 5 значень змінних, причому всі вхідні значення задати дійсними числами у вигляді одновимірних масивів.

5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки, які повинні охоплювати різноманітні сполучення вхідних даних, на які програма повинна надавати вірну відповідь. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі. Точність розрахунків така ж, як і у лаб. роботі 6.

6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.

7. У протоколі по лабораторній роботі для першого і другого способів передачі параметрів поруч з відповідними командами у лістингу відобразити в графічному вигляді стани стека при зверненні до процедур, виконання у них команд та повернення з процедур до основної програми.

8. Зробити висновки по лабораторній роботі.

**Виконання роботи**

Формула для розрахунку за варіантом: (-2 \* c - d \* 82) / tg(a / 4 - b)

Розрахунки

1. Знаменник і чисельник > 0

a = -10,3  
 b = 6.8  
 c = 2.1

d = -0.6

(-2 \* 2.1 - (-0.6) \* 82) / tg(-10.3 / 4 – 6.8) = 45 / tg(-9,375) = 45 / 0,0498191 = 903.268024

1. Чисельник < 0, а знаменник > 0

a = -2.8

b = 2.0

c = 4.5

d = 0.9

(-2 \* 4.5 - (0.9) \* 82) / tg(-2.8 / 4 – 2.0) = 45 / tg(-2,7) = -82.8 / 0,472728 = -175.153577

1. Чисельник > 0, а знаменник < 0

a = 1,6

b = 4.3

c = -3.8

d = -0,7

(-2 \* -3.8 - (-0,7) \* 82) / tg(1,6 / 4 – 4.3) = 65/ tg(-3,9) = 65/-0,947425 = 68,6067965

1. Знаменник і чисельник < 0

a = 5.5

b = 2.2

c = 4.1

d = 2.7

(-2 \* 4.1 - (2.7) \* 82) / tg(5.5 / 4 – 2.2) = -229.6/ tg(-0.825) = -229.6 / -1,08251 = 212,099657

1. Знаменник = 0

a = 4.0

b = 1.0

c = 2.5

d = 3,75

(-2 \* 2.5 - (3,75) \* 82) / tg(4.0 / 4 –1.0) = -312,5/ tg(0) = -312,5 / 0

1. Порушення області визначення

a = 12.56637

b = 1.8

c = 1.8

d = 2.4

(-2 \* 1.8 - (2.4) \* 82) / tg(12.56637 / 4 –1.8) = -200.4/ tg(pi/2)

1. Порушення області визначення

a = 6.28319

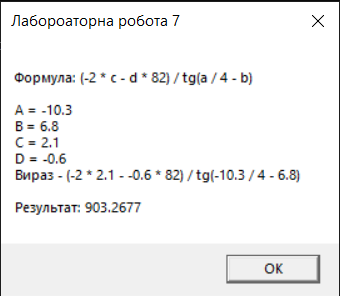
b = 0.0

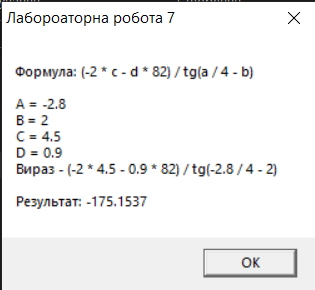
c = 2,7

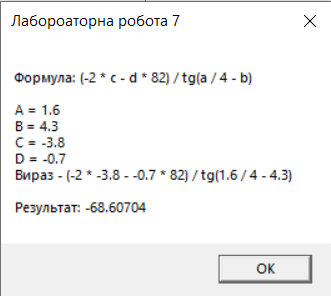
d = 5,3

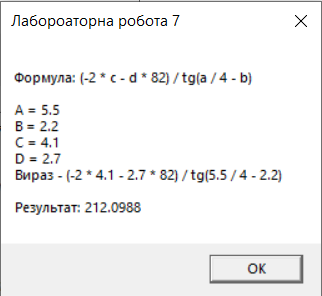
(-2 \* 2,7 - (5,3) \* 82) / tg(6.28319 / 4 - 0) = -440 / tg(pi/2)

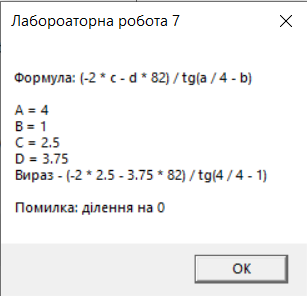
**Скріншоти програми**

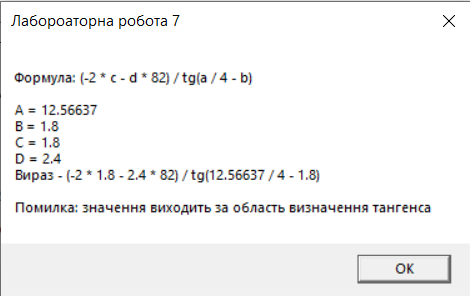


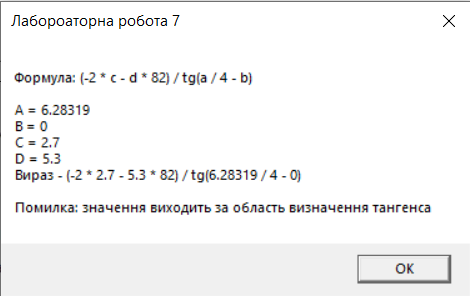












**Лістинг**

**7-2-IM-31-Bulakh.asm**

.386

.model flat, stdcall

option casemap:none

include \masm32\include\windows.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\dialogs.inc

include \masm32\include\masm32.inc

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\masm32.lib

public BohdanBulakh4Const, BohdanBulakhA, BohdanBulakhB

extern BohdabBulakhCalcDenominator:proto

.const

BohdanBulakhMinus2Const dq -2.0

BohdanBulakh4Const dq 4.0

BohdanBulakhConstPi dq 3.141592

BohdanBulakh82Const dq 82.0

BohdanBulakhConstHalfPi dq 1.570796

.data

BohdanBulakhFromulaInfo db "Формула: (-2 \* c - d \* 82) / tg(a / 4 - b)", 10, 10,

"A = %s", 10,

"B = %s", 10,

"C = %s", 10,

"D = %s", 10,

"Вираз - (-2 \* %s - %s \* 82) / tg(%s / 4 - %s)", 10, 10,

"%s", 0

BohdanBulakhResultStrText db "Результат: %s", 0

BohdanBulakhA dq -10.3, -2.8, 1.6, 5.5, 4.0, 12.56637, 6.28319

BohdanBulakhB dq 6.8, 2.0, 4.3, 2.2, 1.0, 1.8, 0.0

BohdanBulakh0DivError db "Помилка: ділення на 0", 0

BohdanBulakhMessageBoxTitle db "Лабороаторна робота 7", 0

BohdanBulakhTanError db "Помилка: значення виходить за область визначення тангенса", 0

BohdanBulakhCurrentStrC db 128 dup (?)

BohdanBulakhCurrentStrA db 128 dup (?)

BohdanBulakhCurrentStrB db 128 dup (?)

BohdanBulakhCurrentStrD db 128 dup (?)

BohdanBulakhStrResultNum db 128 dup (?)

BohdanBulakhC dq 2.1, 4.5, -3.8, 4.1, 2.5, 1.8, 2.7

BohdanBulakhD dq -0.6, 0.9, -0.7, 2.7, 3.75, 2.4, 5.3

BohdanBulakhMessageBoxContent db 256 dup (?)

BohdanBulakhSuccessResultText db 256 dup (?)

BohdanBulakhArgOfTg dt ?

BohdanBulakhResultNumerator dt ?

BohdanBulakhResultDenominator dt ?

BohdanBulakhFirstMulNumer dt ?

BohdanBulakhSecondMulDenom dt ?

BohdanBulakhRem dt ?

BohdanBulakhFloatResult dq ?

BohdanBulakhValidateTanArg macro

fld BohdanBulakhA[esi \* 8]

fdiv BohdanBulakh4Const

fld BohdanBulakhB[esi \* 8]

fsub

fstp BohdanBulakhArgOfTg

fld BohdanBulakhArgOfTg

fldz

fcom

fnstsw ax

sahf

jz BohdanBulakh0Error

fld BohdanBulakhArgOfTg

fsub BohdanBulakhConstHalfPi

fld BohdanBulakhConstPi

fprem

fistp DWord ptr [BohdanBulakhRem]

cmp DWORD ptr [BohdanBulakhRem], 0

je BohdanBulakhTanShowError

endm

.code

BohdanBulakhCalcFirstPart proc

fld qword ptr [ebx]

fmul BohdanBulakhMinus2Const

fstp BohdanBulakhFirstMulNumer

ret[4]

BohdanBulakhCalcFirstPart endp

BohdanBulakhCalcSecondPart proc

push ebp[7]

mov ebp, esp

mov edx, [ebp+8]

fld qword ptr [edx]

fmul BohdanBulakh82Const

fstp BohdanBulakhSecondMulDenom

pop ebp[8]

ret 4[9]

BohdanBulakhCalcSecondPart endp

lab:

mov esi, 0

BohdanBulakhLoop:

invoke FloatToStr, BohdanBulakhA[esi \* 8],

addr BohdanBulakhCurrentStrA

invoke FloatToStr, BohdanBulakhB[esi \* 8],

addr BohdanBulakhCurrentStrB

invoke FloatToStr,

BohdanBulakhC[esi \* 8],

addr BohdanBulakhCurrentStrC

invoke FloatToStr,

BohdanBulakhD[esi \* 8], addr BohdanBulakhCurrentStrD

finit

lea ebx, BohdanBulakhC[esi\*8]

call BohdanBulakhCalcFirstPart[3]

lea eax, BohdanBulakhD[esi\*8]

push eax[5]

call BohdanBulakhCalcSecondPart[6]

fld BohdanBulakhFirstMulNumer

fld BohdanBulakhSecondMulDenom

fsub

fstp BohdanBulakhResultNumerator

BohdanBulakhValidateTanArg

call BohdabBulakhCalcDenominator[1]

fstp st(0)

fstp BohdanBulakhResultDenominator

fld BohdanBulakhResultNumerator

fld BohdanBulakhResultDenominator

fdiv

fstp BohdanBulakhFloatResult

invoke FloatToStr, BohdanBulakhFloatResult, addr BohdanBulakhStrResultNum

invoke wsprintf, addr BohdanBulakhSuccessResultText, addr BohdanBulakhResultStrText, addr BohdanBulakhStrResultNum

invoke wsprintf, addr BohdanBulakhMessageBoxContent, addr BohdanBulakhFromulaInfo,

addr BohdanBulakhCurrentStrA, addr BohdanBulakhCurrentStrB,

addr BohdanBulakhCurrentStrC, addr BohdanBulakhCurrentStrD,

addr BohdanBulakhCurrentStrC,

addr BohdanBulakhCurrentStrD, addr BohdanBulakhCurrentStrA, addr BohdanBulakhCurrentStrB,

addr BohdanBulakhSuccessResultText

jmp BohdanBulakhShowResult

BohdanBulakh0Error:

invoke wsprintf, addr BohdanBulakhMessageBoxContent,

addr BohdanBulakhFromulaInfo,

addr BohdanBulakhCurrentStrA,

addr BohdanBulakhCurrentStrB,

addr BohdanBulakhCurrentStrC, addr BohdanBulakhCurrentStrD,

addr BohdanBulakhCurrentStrC, addr BohdanBulakhCurrentStrD,

addr BohdanBulakhCurrentStrA, addr BohdanBulakhCurrentStrB,

addr BohdanBulakh0DivError

jmp BohdanBulakhShowResult

BohdanBulakhShowResult:

invoke MessageBox, 0, addr BohdanBulakhMessageBoxContent, addr BohdanBulakhMessageBoxTitle, 0

jmp BohdanBulakhCheckIteration

BohdanBulakhTanShowError:

invoke wsprintf, addr BohdanBulakhMessageBoxContent, addr BohdanBulakhFromulaInfo,

addr BohdanBulakhCurrentStrA, addr BohdanBulakhCurrentStrB,

addr BohdanBulakhCurrentStrC, addr BohdanBulakhCurrentStrD,

addr BohdanBulakhCurrentStrC, addr BohdanBulakhCurrentStrD,

addr BohdanBulakhCurrentStrA, addr BohdanBulakhCurrentStrB,

addr BohdanBulakhTanError

jmp BohdanBulakhShowResult

BohdanBulakhCheckIteration:

inc esi

.if esi < 7

jmp BohdanBulakhLoop

.else

invoke ExitProcess, 0

.endif

end lab

**7-2-IM-31-Bulakh-PUB.asm**

.386

.model flat, stdcall

option casemap:none

public BohdabBulakhCalcDenominator

extern BohdanBulakh4Const:qword, BohdanBulakhA:qword, BohdanBulakhB:qword

.code

BohdabBulakhCalcDenominator proc

fld BohdanBulakhA[esi \* 8]

fdiv BohdanBulakh4Const

fld BohdanBulakhB[esi \* 8]

fsub

fptan

ret[2]

BohdabBulakhCalcDenominator endp

end

**7-2-IM-31-Bulakh.bat**

@echo off

ML /c /coff "7-2-IM-31-Bulakh.asm"

ML /c /coff "7-2-IM-31-Bulakh-PUB.asm"

link32 /subsystem:windows "7-2-IM-31-Bulakh.obj" "7-2-IM-31-Bulakh-PUB.obj"

7-2-IM-31-Bulakh.exe

**Стани стеку**

|  |  |
| --- | --- |
| 1. Виклик BohdabBulakhCalcDenominator | |
| max |  |
| ... | ... |
| EIP | Адреса повернення |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 2. Повернення з BohdabBulakhCalcDenominator | |
| max |  |
| ... | ... |
|  |  |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 3. Виклик BohdanBulakhCalcFirstPart | |
| max |  |
| ... | ... |
| EIP | Адреса повернення |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 4. Повернення з BohdanBulakhCalcFirstPart | |
| max |  |
| ... | ... |
|  |  |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 5. Параметр eax (адреса d) передається у стек | |
| max |  |
| ... | ... |
| eax | адреса d |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 6. Виклик BohdanBulakhCalcSecondPart | |
| max |  |
| ... | ... |
| EIP | Адреса повернення |
| eax | адреса d |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 7. Збереження ebp у процедурі BohdanBulakhCalcSecondPart | |
| max |  |
| ... | ... |
| ebp | Значення регістра EBP |
| EIP | адреса повернення |
| eax | адреса d |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 8. Відновлення ebp після виконання процедури | |
| max |  |
| ... | ... |
| EIP | Адреса повернення |
| eax | адреса d |
|  |  |
| min |  |

|  |  |
| --- | --- |
| 9. Повернення з BohdanBulakhCalcSecondPart | |
| max |  |
| ... | ... |
|  |  |
|  |  |
| min |  |

**Висновки**

У ході виконання лабораторної роботи було розроблено програму на мові асемблера для обчислення формули з використанням трьох методів передачі параметрів: через регістри, стек та директиву EXTRN/PUBLIC. Для першого доданка чисельника параметр c передавався через регістр EBX, що забезпечило швидкість виконання. Другий доданок обчислювався через передачу адреси d у стек, що дозволило гнучко працювати з даними. Знаменник було винесено в окремий файл з використанням EXTRN/PUBLIC, що підкреслило переваги модульності.

Програма коректно обробляє помилки: ділення на нуль та випадки, коли аргумент тангенса наближається до π/2. Результати роботи збігаються з контрольними розрахунками з точністю до 6 знаків після коми. Під час аналізу станів стеку встановлено, що передача параметрів через стек потребує збереження адреси повернення (EIP) та використання EBP для доступу до даних, а команда ret 4 забезпечує коректне очищення стеку.