**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 №5**

з дисципліни «Системне програмування»

на тему «Арифметичні і логічні операції з цілими числами. Масиви.»

Виконав: Перевірив:

Студент 2 курсу ФІОТ доц. Павлов В. Г.

групи ІМ-22

Довженко Антон Андрійович

номер в списку групи: 9

Київ 2024

**Мета роботи:** Вивчення арифметичних і логічних команд Асемблера і здобуття навиків виконання розрахунків з елементами масивів.

**Контрольні приклади:**

1. І чисельник і знаменник більші за нуль:

a = 2, b = 2, c =10

(21 - 2\*10/4)/(1 + 10/2 + 2) = (21 - 5) / (1 + 5 + 2) = 16/8 = 2 – це у нас парний результат, тому ділимо його на 2: 2 / 2 =1

2. Чисельник більший за нуль, а знаменник менший:

a = -4, b = -10, c = 24

(21 – (-4) \* 24/4)/( 1 + 24/(-4) + (-10)) = (21 – (-24) ) / (1 – 6 - 10) = 45 / (-15) = -3 – непарний результат, тому множимо на 5: -3 \* 5 = -15

3. Чисельник менший за нуль, а знаменник більший:

a = 3, b = -14, c = 48

(21 – 3 \* 48/4)/( 1 + 48/3 - 14) = (21 – 36)/(1 + 16 - 14) = (-15)/3 = -5 – результат непарний, тому множимо на 5: -5 \* 5 = -25

4. І чисельник і знаменник менші за нуль:

a = 7, b = 12, c = 28

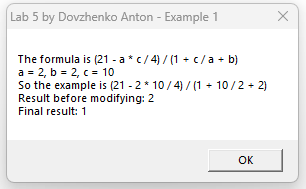
(21 – 7\*28/4)/(1 + 28/7 - 12) = (21 - 49)/ (1 + 4 - 12) = (-28)/(-7) = 4 – цей результат у нас парний, тому ми його ділимо на 2: 4 / 2 = 2

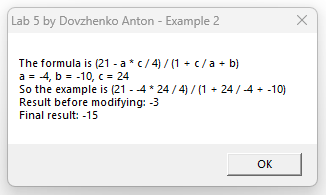
5. Знаменник дорівнює 0:

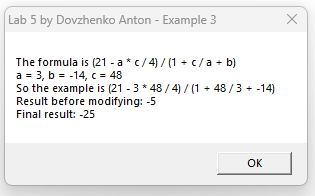
a = 4, b = -3, c = 8

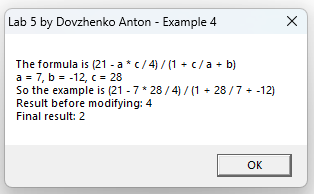
(21 - 4\*8/4)/( 1 + 8/4 - 3) = (21 - 8) / (1 + 2 - 3)= 13/0

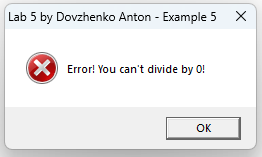
**Скріншоти роботи програми**











Отже, в результаті порівняння контрольних розрахунків з результатами, що було виведеною програмою, можна сказати, що в усіх випадках програма правильно обраховує результат, а при діленні на 0 вона виводить помилку та вказує в якому прикладі ця помилка.

**Лістинг програми**

.386

.model flat, stdcall

option casemap :none

include \masm32\include\masm32rt.inc

.data

DovzhenkoBasicTitle db "Lab 5 by Dovzhenko Anton - Example %d", 0

DovzhenkoDivByZeroErrorMsg db "Error! You can't divide by 0!", 0

DovzhenkoResultsWin db "The formula is (21 - a \* c / 4) / (1 + c / a + b)", 13,

"a = %d, b = %d, c = %d", 13,

"So the example is (21 - %d \* %d / 4) / (1 + %d / %d + %d)", 13,

"Result before modifying: %d", 13,

"Final result: %d", 0

DovzhenkoFourKm dd 4

.data

DovzhenkoA dd 2, -4, 3, 7, 4

DovzhenkoB dd 2, -10, -14, -12, -3

DovzhenkoC dd 10, 24, 48, 28, 8

.code

start:

main proc

LOCAL DovzhenkoLpIndx: DWORD

LOCAL DovzhenkoTitle[100]: BYTE

LOCAL DovzhenkoError[100]: BYTE

LOCAL DovzhenkoResultStr[256]: BYTE

LOCAL DovzhenkoIntrmdRes: DWORD

LOCAL DovzhenkoFinalRes: DWORD

LOCAL DovzhenkoPAVal: DWORD, DovzhenkoPBVal: DWORD, DovzhenkoPCVal: DWORD

LOCAL DovzhenkoNumZm: DWORD, DovzhenkoDenumZN: DWORD

mov DovzhenkoLpIndx, 0

DovzhenkoCalcLoop:

mov eax, DovzhenkoLpIndx

inc eax ; Increment index

invoke wsprintf, addr DovzhenkoTitle, addr DovzhenkoBasicTitle , eax ; Format the window title

invoke wsprintf, addr DovzhenkoError, addr DovzhenkoDivByZeroErrorMsg ; Format the error message

mov eax, DovzhenkoLpIndx

shl eax, 2

lea ebx, [DovzhenkoA]

add ebx, eax

mov ecx, [ebx]

mov DovzhenkoPAVal, ecx

lea ebx, [DovzhenkoB]

add ebx, eax

mov ecx, [ebx]

mov DovzhenkoPBVal, ecx

lea ebx, [DovzhenkoC]

add ebx, eax

mov ecx, [ebx]

mov DovzhenkoPCVal, ecx

; Multiplication

mov eax, DovzhenkoPAVal

imul eax, DovzhenkoPCVal

mov ecx, [DovzhenkoFourKm]

or ecx, ecx

jz DovzhenkoDivByZeroErr

cdq

idiv ecx

mov ecx, 21

sub ecx, eax

mov DovzhenkoNumZm, ecx

; Division calculation and zero-check

mov eax, DovzhenkoPCVal

or eax, eax

jz DovzhenkoDivByZeroErr

cdq

mov ecx, DovzhenkoPAVal

idiv ecx

add eax, DovzhenkoPBVal

inc eax

or eax, eax

jz DovzhenkoDivByZeroErr

mov DovzhenkoDenumZN, eax

mov eax, DovzhenkoNumZm

or eax, eax

jz DovzhenkoDivByZeroErr

cdq

idiv DovzhenkoDenumZN

mov DovzhenkoIntrmdRes, eax

; Check if the result is even or odd

mov eax, DovzhenkoIntrmdRes

test eax, 1

jz DovzhenkoEvenIntResult

imul eax, 5

jmp DovzhenkoSetFinalRes

DovzhenkoEvenIntResult:

sar eax, 1

DovzhenkoSetFinalRes:

mov DovzhenkoFinalRes, eax

invoke wsprintf, addr DovzhenkoResultStr, addr DovzhenkoResultsWin, DovzhenkoPAVal, DovzhenkoPBVal, DovzhenkoPCVal, DovzhenkoPAVal, DovzhenkoPCVal, DovzhenkoPCVal, DovzhenkoPAVal, DovzhenkoPBVal, DovzhenkoIntrmdRes, DovzhenkoFinalRes

invoke MessageBox, NULL, addr DovzhenkoResultStr, addr DovzhenkoTitle, MB\_OK

add DovzhenkoLpIndx, 1

cmp DovzhenkoLpIndx, 5

jl DovzhenkoCalcLoop

jmp DovzhenkoExitProcessLbl

DovzhenkoDivByZeroErr:

invoke MessageBox, NULL, addr DovzhenkoError, addr DovzhenkoTitle, MB\_ICONERROR or MB\_OK

jmp DovzhenkoExitProcessLbl

DovzhenkoExitProcessLbl:

invoke ExitProcess, 0

main endp

end start

**Висновки:**

Отже, при виконанні лабораторної роботи я ознайомився та засвоїв арифметичні та логічні команди мови асемблера (MASM32) та використав їх на практиці в програмі, що робить обчислення по формулі. У результаті порівнянь контрольних обчислень з результатами програми я можу дійти до висновка, що програма була написана правильно і у всіх випадках (які залежать від знаків чисельника та знаменника) обраховує програма правильно. Також вона ефективно перевіряє чи є у нас ділення на 0, і якщо ж воно є, то програма виводить помилку, повідомляючи, що таке неприпустимо.