**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 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 арифметичних і логічних команд Асемблера і

здобуття навиків виконання розрахунків з елементами масивів.

Завдання за варіантом:

номер у списку групи: 20, тож 20 варіант: (c - 33 + b/4)/(a\*c/b - 1), або це можна записати так: .

**Контрольні приклади:**

1. Чисельник і знаменник більші за нуль.

a = 7, b = 32, c = 64.

. Цей результат є непарним, тому ми додатково множимо його на 5: 3 \* 5 = 15.

Таким чином:

Проміжний результат = 3

Результат після додаткової дії = 15

2. Чисельник та знаменник менші за нуль.

a = 8, b = 20, c = -40.

. Цей результат є парним, тому ми додатково ділимо його на 2: 4 / 2 = 2.

Таким чином:

Проміжний результат = 4

Результат після додаткової дії = 2

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

a = -7, b = 48, c = 96.

. Цей результат є непарним, тому ми додатково множимо його на 5: -5 \* 5 = -25.

Таким чином:

Проміжний результат = -5

Результат після додаткової дії = -25

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

a = -21, b = 28, c = -56.

. Цей результат є парним, тому ми додатково ділимо його на 2: -2 / 2 = -1.

Таким чином:

Проміжний результат = -2

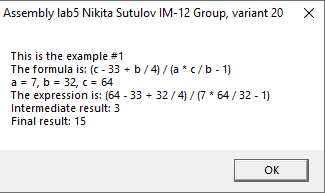
Результат після додаткової дії = -1

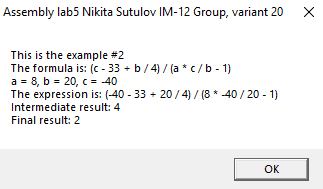
5. Знаменник дорівнює нулю.

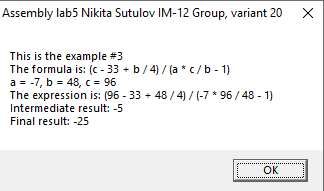
a = 2, b = 4, c = 2

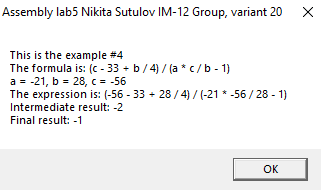
.

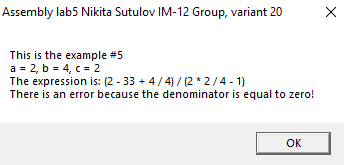
**Демонстрація роботи програми**











В результаті порівняння контрольних розрахунків з результатами виконання програми видно, що дані збігаються, отже програма 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).

**Лістинг програми**

.386

.model flat, stdcall

option casemap :none

include \masm32\include\masm32rt.inc

.data

; arrays

SutulovAArray dd 7, 8, -7, -21, 2

SutulovBArray dd 32, 20, 48, 28, 4

SutulovCArray dd 64, -40, 96, -56, 2

; window caption

SutulovWindowCaption db "Assembly lab5 Nikita Sutulov IM-12 Group, variant 20", 0

; window text template

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

SutulovFormula db "The formula is: (c - 33 + b / 4) / (a \* c / b - 1)", 0

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

SutulovExpressionForm db "The expression is: (%d - 33 + %d / 4) / (%d \* %d / %d - 1)", 0

SutulovIntermediateResultForm db "Intermediate result: %d", 0

SutulovFinalResultForm db "Final result: %d", 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", 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 64 dup (?)

SutulovIntermediateResultBuffer db 64 dup (?)

SutulovFinalResultBuffer db 64 dup (?)

SutulovFinalBuffer db 256 dup (?)

SutulovIntermediateResult dd ?

SutulovDenominator dd ?

SutulovFinalResult dd ?

SutulovCurrentA dd ?

SutulovCurrentB dd ?

SutulovCurrentC dd ?

.code

main:

mov edi, 0 ; counter for indexes

mov esi, 1 ; counter for example numbers

loopToIterateThroughArrays:

cmp edi, 5

je exitLoop

mov eax, SutulovAArray[edi \* 4]

mov ebx, SutulovBArray[edi \* 4]

mov ecx, SutulovCArray[edi \* 4]

mov SutulovCurrentA, eax

mov SutulovCurrentB, ebx

mov SutulovCurrentC, ecx

; a \* c

imul eax, ecx

; a \* c / b

cdq

idiv ebx

; a \* c / b - 1

sub eax, 1

cmp eax, 0

je denominatorZero

mov SutulovDenominator, eax

; b / 4

mov eax, SutulovCurrentB

mov ecx, 4

cdq

idiv ecx

; c - 33 + b / 4

add eax, SutulovCurrentC

sub eax, 33

; (c - 33 + b / 4) / (a \* c / b - 1)

mov ecx, SutulovDenominator

cdq

idiv ecx

; saving the intermediate result

mov SutulovIntermediateResult, eax

; checking if the result is odd or even

test eax, 1

jz evenResult

; if the number is odd, multiply by 5

mov ecx, 5

imul eax, 5

mov SutulovFinalResult, eax

normalWindow:

; making the window text

invoke wsprintf, offset SutulovExampleBuffer,

offset SutulovExampleForm, esi

invoke wsprintf, offset SutulovCurrentValuesBuffer,

offset SutulovCurrentValuesForm,

SutulovCurrentA, SutulovCurrentB, SutulovCurrentC

invoke wsprintf, offset SutulovExpressionBuffer,

offset SutulovExpressionForm,

SutulovCurrentC, SutulovCurrentB, SutulovCurrentA,

SutulovCurrentC, SutulovCurrentB

invoke wsprintf, offset SutulovIntermediateResultBuffer,

offset SutulovIntermediateResultForm, SutulovIntermediateResult

invoke wsprintf, offset SutulovFinalResultBuffer,

offset SutulovFinalResultForm, SutulovFinalResult

invoke wsprintf, offset SutulovFinalBuffer, offset SutulovFinalForm,

offset SutulovExampleBuffer, offset SutulovFormula,

offset SutulovCurrentValuesBuffer, offset SutulovExpressionBuffer,

offset SutulovIntermediateResultBuffer, offset SutulovFinalResultBuffer

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

inc edi

inc esi

jmp loopToIterateThroughArrays

denominatorZero:

invoke wsprintf, offset SutulovExampleBuffer,

offset SutulovExampleForm, esi

invoke wsprintf, offset SutulovCurrentValuesBuffer,

offset SutulovCurrentValuesForm,

SutulovCurrentA, SutulovCurrentB, SutulovCurrentC

invoke wsprintf, offset SutulovExpressionBuffer,

offset SutulovExpressionForm,

SutulovCurrentC, SutulovCurrentB, SutulovCurrentA,

SutulovCurrentC, SutulovCurrentB

invoke wsprintf, offset SutulovIntermediateResultBuffer,

offset SutulovErrorMessageForm, offset SutulovErrorMessage

invoke wsprintf, offset SutulovFinalBuffer, offset SutulovErrorForm,

offset SutulovExampleBuffer,

offset SutulovCurrentValuesBuffer,

offset SutulovExpressionBuffer,

offset SutulovIntermediateResultBuffer

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

inc edi

inc esi

jmp loopToIterateThroughArrays

evenResult:

mov ecx, 2

cdq

; if the number is even, divide by two

idiv ecx

mov SutulovFinalResult, eax

jmp normalWindow

exitLoop:

invoke ExitProcess, 0

end main

**Висновки**

У результаті порівняння контрольних розрахунків із результатами виконання програми для всіх передбачених лабораторною роботою сценаріїв:

* 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 як проміжні, так і остаточні результати обчислень співпадають, і випадок, коли знаменник дорівнює 0, обробляється коректно. Таким чином, роблю висновок, що програму написано коректно.

У ході виконання цієї лабораторної роботи я познайомився та вивчив арифметичні та логічні команди в мові асемблера MASM32 (присвоєння mov, додавання add, віднімання sub, цілочисельного множення imul, цілочисельного ділення idiv, порівняння cmp тощо), а також здобув навички здобув навички здійснення розрахунків з елементами одновимірних масивів.