**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 групи ІМ-22 доц. Павлов В. Г.

Тимофеєв Даниіл Костянтинович

номер в списку групи: 23

Київ 2024

**Мета роботи:**

Вивчення арифметичних і логічних команд Асемблера і здобуття навиків виконання розрахунків з елементами масивів.

**Порядок виконання роботи:**

1. Вивчити арифметичні і логічні команди цілочисельної арифметики [1, 2].
2. Розробити програму на мові Асемблер, в якій згідно з індивідуальним варіантом завдання (таблиця. 1) виконуються обчислення значення арифметичного виразу з подальшим виводом результату\* у віконному інтерфейсі.
3. Для всіх варіантів: в разі парного результату він перед виводом додатково ділиться на 2, а в разі непарного – результат додатково умножається на 5.
4. Розрахунки (п. 2, 3) повторити в програмі для 5 значень змінних\*\*, причому всі вихідні значення задати цілими числами у вигляді одновимірних масивів.
5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі.
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. Зробити висновки по лабораторній роботі.

**Пункт 5.** Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі.

**Контрольні розрахунки**

Номер у списку групи = 23, тож 23 - 20 = 3 варіант для цієї лабораторної роботи. Формула (c/d + 3\*a/2)/(c - a + 1). Розгляньмо п’ять випадків:

* чисельник і знаменник більші за нуль;
* чисельник більше за нуль, але знаменник менший за нуль;
* чисельник менше за нуль, а знаменник більший за нуль;
* чисельник і знаменник менші за нуль;
* знаменник дорівнює нулю

1. Чисельник і знаменник більші за нуль:

числа: a = 4, c = 6, d = 2

формула: (6/2 + 3\*4/2)/(6 - 4 + 1) = (3 + 6)/3 = 9/3 = 3. Оскільки число 3 є непарним => множимо на 5, і виходить 3 \* 5 = 15

1. Чисельник більше за нуль, але знаменник менший за нуль:

числа: a = 10, c = 7, d = 7

формула: (7/7 + 3\*10/2)/(7 - 10 + 1) = (1 + 3\*5)/(-2) = (1 + 15)/(-2) = 16/(-2) = -8. Оскільки число (-8) є парним, то ми його ділимо на 2, тож результат = (-8)/2 = -4

1. Чисельник менше за нуль, а знаменник більший за нуль:

числа: a = -2, c = 24, d = -1

формула: (24/(-1) + 3\*(-2)/2)/(24 - (-2) + 1) = (-24 + 3\*(-1))/(24 + 2 + 1) = (-24 - 3)/(27) = -27/27 = -1. Оскільки число (-1) є непарним => множимо на 5, і виходить (-1) \* 5 = -5

1. Чисельник і знаменник менші за нуль

числа: a = -25, c = -27, d = 6

формула: (-27/6 + 3\*(-25)/2)/(-27 - (-25) + 1) = ((-9)/2 + 3\*(-25)/2)/(-27 + 25 +1) = ((-9)/2 - 75/2)/(-1) = (-42) / (-1) = 42. Оскільки число 42 є парним, то ми його ділимо на 2, тож результат = 42/2 = 21

1. Знаменник дорівнює нулю:

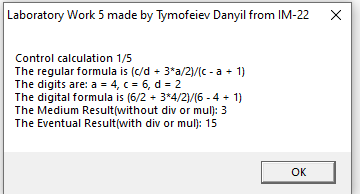
числа: a = 38, c = 37, d = 43

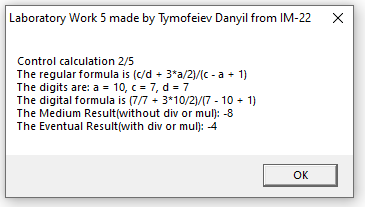
формула: (37/43 + 3\*38/2)/(37 - 38 + 1) = (37/43 + 3\*19) / 0 = (37/43 +57) / 0 = (2488/43)/0. На 0 ділити не можемо => у програмі кидаємо помилку

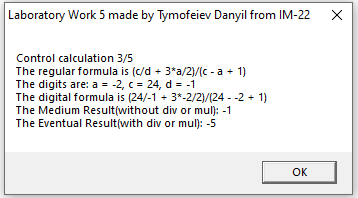
**Пункт 6.** Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.

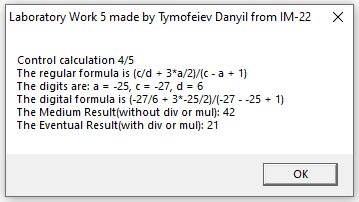
**Скріншоти виконання роботи**

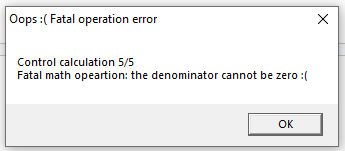
Порядок обрахунків такий самий, що і в контрольних розрахунках:











Що ж, було продемонстровано коректні обрахунки для всіх вище перерахованих випадків, які збігаються з контрольними розрахунками. Для нуля в знаменнику, як бачимо, кидається помилка.

**Лістинг розробленої програми**

.686

.model flat, stdcall

option casemap:none

include \masm32\include\masm32rt.inc

include \masm32\include\dialogs.inc

.data?

TymofeievSummaryBuffer db 256 dup (?)

TymofeievSummaryBufferForTempValues db 256 dup (?)

TymofeievMediumResult dd ?

TymofeievEventualResult dd ?

TymofeievTemporaryAState dd ?

TymofeievRoutineAState dd ?

TymofeievTemporaryDState dd ?

TymofeievRoutineDState dd ?

TymofeievTemporaryCState dd ?

TymofeievRoutineCState dd ?

.data

ControlCalculationsCount dd 1

ArrayStartPosition dd 0

TymofeievArrayForAValues dd 4, 10, -2, -25, 38

TymofeievArrayForCValues dd 6, 7, 24, -27, 37

TymofeievArrayForDValues dd 2, 7, -1, 6, 43

TymofeievFatalIOperationCaption db "Oops :( Fatal operation error", 0

TymofeievFatalIOperationBody db "Fatal math opeartion: the denominator cannot be zero :(", 0

TymofeievValidLabCaption db "Laboratory Work 5 made by Tymofeiev Danyil from IM-22", 0

TymofeivVariantMessgBoxInfo db "Control calculation %d/5", 13,

"The regular formula is (c/d + 3\*a/2)/(c - a + 1)", 13,

"The digits are: a = %d, c = %d, d = %d", 13,

"The digital formula is (%d/%d + 3\*%d/2)/(%d - %d + 1)", 13,

"The Medium Result(without div or mul): %d", 13,

"The Eventual Result(with div or mul): %d", 0

TymofeivVariantInvalidMessgBoxInfo db "Control calculation %d/5", 13,

"Fatal math opeartion: the denominator cannot be zero :(", 0

.code

TymofeievCheckTemporaryStatesEvenOrOdd PROC

push ebp

mov ebp, esp

mov eax, [ebp + 8]

test eax, 1

jz EvenNumber

jmp OddNumber

EvenNumber:

mov eax, 0

jmp EndCheck

OddNumber:

mov eax, 1

jmp EndCheck

EndCheck:

pop ebp

ret 4

TymofeievCheckTemporaryStatesEvenOrOdd ENDP

start:

mov ArrayStartPosition, 0

mov ControlCalculationsCount, 1

TymofeievCountTheFormulaLoop:

mov eax, ArrayStartPosition

; Multiply by 4

mov edx, 4

mul edx

; Loading TymofeievRoutineAState from TymofeievArrayForAValues

mov ecx, [TymofeievArrayForAValues + eax]

mov TymofeievRoutineAState, ecx

; Loading TymofeievRoutineCState from TymofeievArrayForCValues

mov ecx, [TymofeievArrayForCValues + eax]

mov TymofeievRoutineCState, ecx

; Loading TymofeievRoutineDState from the TymofeievArrayForDValues array

mov ecx, [TymofeievArrayForDValues + eax]

mov TymofeievRoutineDState, ecx

; Numerator calculation

mov eax, TymofeievRoutineCState ; c

cdq

mov ebx, TymofeievRoutineDState ; d

; Check for division by zero in the denominator (d)

cmp ebx, 0

je TymofeievOutputInvalidMSGBoxLoop

idiv ebx ; c / d

mov ecx, eax ; Save result c / d in ecx

mov eax, TymofeievRoutineAState

imul eax, 3

sar eax, 1

add eax, ecx

; Denominator calculation

mov ecx, TymofeievRoutineCState ; c

sub ecx, TymofeievRoutineAState ; c - a

inc ecx ; c - a + 1

; Check for division by zero in the denominator

cmp ecx, 0

je TymofeievOutputInvalidMSGBoxLoop

; Final division

cdq

idiv ecx

mov TymofeievMediumResult, eax

; even or odd?

mov eax, TymofeievMediumResult

; even or odd?

test eax, 1

jz TymofeievCheckPairOfNumbersLoop

; is odd => \*5

jmp TymofeievCheckOddNumbersLoop

; is odd => \*5

TymofeievCheckOddNumbersLoop:

imul eax, 5

jmp TymofeievOutputNormalMSGBoxLoop

; is even => /2

TymofeievCheckPairOfNumbersLoop:

mov ebx, 2

cdq

idiv ebx

jmp TymofeievOutputNormalMSGBoxLoop

TymofeievOutputNormalMSGBoxLoop:

invoke wsprintf, offset TymofeievSummaryBuffer, offset TymofeivVariantMessgBoxInfo,

ControlCalculationsCount, TymofeievRoutineAState, TymofeievRoutineCState, TymofeievRoutineDState,

TymofeievRoutineCState, TymofeievRoutineDState, TymofeievRoutineAState,

TymofeievRoutineCState, TymofeievRoutineAState, TymofeievMediumResult, eax

invoke MessageBox, NULL, offset TymofeievSummaryBuffer, offset TymofeievValidLabCaption, MB\_OK

inc ArrayStartPosition

inc ControlCalculationsCount

jmp TymofeievLoopCheck

TymofeievLoopCheck:

push ControlCalculationsCount

; Checking whether the number of calculations has exceeded the limit of 6

cmp ControlCalculationsCount, 6

jge TymofeievEscapeLoop ; If it reaches 6 or more, we exit the cycle

; In other cases, we continue the calculation cycle

jmp TymofeievCountTheFormulaLoop

pop ControlCalculationsCount

TymofeievOutputInvalidMSGBoxLoop:

mov edx, 1

; Perform the MessageBox operation

invoke wsprintf,

offset TymofeievSummaryBuffer,

offset TymofeivVariantInvalidMessgBoxInfo,

ControlCalculationsCount

invoke MessageBox, NULL, offset TymofeievSummaryBuffer, offset TymofeievFatalIOperationCaption, MB\_OK

imul edx, edx, 2

test edx, edx

jz TymofeievEscapeLoop

TymofeievEscapeLoop:

invoke ExitProcess, 0

TymofeievDisplayHandlerInvalidOperationProc proc

invoke wsprintf, offset TymofeievSummaryBuffer, offset TymofeievFatalIOperationCaption, ControlCalculationsCount

invoke MessageBox, NULL, offset TymofeievSummaryBuffer, offset TymofeievFatalIOperationCaption, MB\_OK

ret

TymofeievDisplayHandlerInvalidOperationProc endp

end start

**Пункт 7.** Зробити висновки по лабораторній роботі

**Висновки**

Отже, під час виконання лабораторної роботи я вивчив, як користуватися арифметичними та логічними командами на мові Асемблер і здобув навички на практиці з виконання розрахунків з елементами масивів. Перш за все, я познайомився з циклами та масивами на MASM32 та як їх використовувати на прикладі розрахунків з формули. Ба більше, було описано 5 основних випадків для програми: коли чисельник і знаменник більші за нуль; чисельник більше за нуль, але знаменник менший за нуль; чисельник менше за нуль, а знаменник більший за нуль; чисельник і знаменник менші за нуль; знаменник дорівнює нулю. Щодо порівнянь програми та контрольних розрахунків, то можу стверджувати, що як і проміжні числа, підставлені у формулу, та проміжні значення до ділення чи множення (в залежності від парності чисел), так і фінальний результат було отримано правильно, бо програмна реалізація та мануальні розрахунки повністю збігаються. Більш того, написана мною програма викидає повідомлення про помилку, якщо було знайдено, що є ділення на нуль (знаменник = 0), що є правильною поведінкою.