Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. 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-у доц. Павлов В.Г.

групи ІМ-33

Козарезов Кирил Олександрович

номер у списку групи - 10

Київ 2025

**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.
2. Розробити програму на мові Асемблер, в якій згідно з індивідуальним варіантом завдання 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

**Виконання роботи**

Формула для розрахунку за 10 варіантом: (b/c - 24 + a)/(2\*a\*c - 12).

Контрольні розрахунки

1. Чисельник та знаменник > 0

a = 6  
b = 60  
c = 2

(60/2 - 24 + 6)/(2\*6\*2 - 12)= (30 - 24 + 6) / (24 -12) = 12/ 12= 1

Оскільки результат непарний то множимо на 5

1 \* 5 = 5

Проміжний результат = 1

Остаточний результат = 5

1. Чисельник < 0, а знаменник > 0

a = 5

b = 6

c = 2

(6/2 - 24 + 5)/(2\*5\*2 - 12)= (3 - 24 + 5) / (20 -12) = -16/ 8= -2

Оскільки результат парний то ділимо на 2

-2/2=-1

Проміжний результат = -2

Остаточний результат = -1

1. Чисельник > 0, а знаменник < 0

a = 4

b = 40

c = 1

(40/1 - 24 + 4)/(2\*4\*1 - 12)= (40 - 24 + 4) / (8 -12) = 20/ -4= -5

Оскільки результат непарний то множимо на 5

-5\*5=-25

Проміжний результат = -5

Остаточний результат = -25

1. Знаменник і чисельник < 0

a = 2

b = -36

c = 2

(-36/2 - 24 + 2)/(2\*2\*2 - 12)= (-18 - 24 + 2) / (8 -12) = -40/ -4= 10

Оскільки результат парний то ділимо на 2

10/2=5

Проміжний результат = 10

Остаточний результат = 5

1. Знаменник = 0

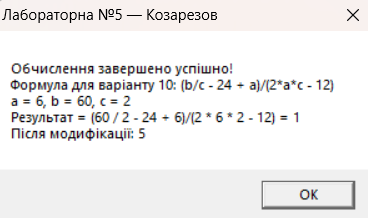
a = 2

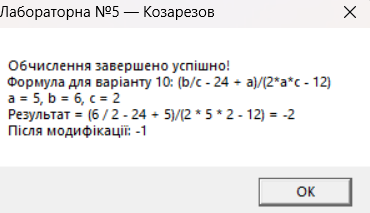
b = 9

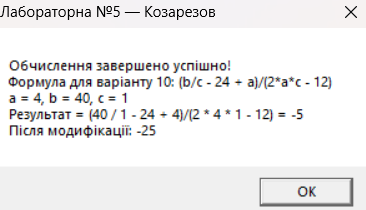
c = 3

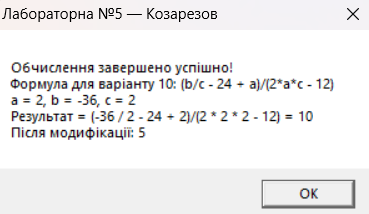
(9/3 - 24 + 2)/(2\*3\*2 - 12)= (3 - 24 + 2) / (12 -12) = -19/ 0

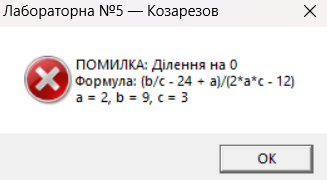
Робота програми











**Лістинг програми**

.386

.model flat, stdcall

option casemap: none

include \masm32\include\dialogs.inc

include \masm32\include\masm32rt.inc

.data?

KozarezovResult dd ?

KozarezovNumerator dd ?

KozarezovDenominator dd ?

KozarezovMsgBuffer db 512 dup(?)

KozarezovTmp dd ?

.data

KozarezovMsgCaption db "Лабораторна №5 — Козарезов",0

KozarezovMsgSuccess db "Обчислення завершено успішно!",13,

"Формула для варіанту 10: (b/c - 24 + a)/(2\*a\*c - 12)",13,

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

"Результат = (%i / %i - 24 + %i)/(2 \* %i \* %i - 12) = %i",13,

"Після модифікації: %i",0

KozarezovMsgFail db "ПОМИЛКА: Ділення на 0",13,

"Формула: (b/c - 24 + a)/(2\*a\*c - 12)",13,

"a = %i, b = %i, c = %i",0

KozarezovArrayA dd 6, 5, 4,2, 2

KozarezovArrayB dd 60, 6, 40, -36, 9

KozarezovArrayC dd 2, 2, 1, 2, 3

.code

start:

mov esi, 0

KozarezovCalcLoop:

cmp esi, 5

jge KozarezovExit

mov eax, KozarezovArrayA[esi\*4]

mov ebx, KozarezovArrayB[esi\*4]

mov ecx, KozarezovArrayC[esi\*4]

mov edx, ecx

cmp ecx, 0

je KozarezovDivZero

; b / c

mov eax, ebx

cdq

idiv ecx

mov ebx, eax ; b/c

mov eax, KozarezovArrayA[esi\*4]

sub ebx, 24

add ebx, eax ; numerator = b/c - 24 + a

mov KozarezovNumerator, ebx

; denominator = 2 \* a \* c - 12

mov eax, KozarezovArrayA[esi\*4]

imul eax, 2

imul eax, KozarezovArrayC[esi\*4]

sub eax, 12

cmp eax, 0

je KozarezovDivZero

mov KozarezovDenominator, eax

mov eax, KozarezovNumerator

mov ebx, KozarezovDenominator

cdq

idiv ebx

mov KozarezovResult, eax

; модифікація результату

test eax, 1

jnz KozarezovOdd

mov eax, KozarezovResult

cdq

mov ebx, 2

idiv ebx

jmp KozarezovShowResult

KozarezovOdd:

mov eax, KozarezovResult

imul eax, 5

KozarezovShowResult:

invoke wsprintf, addr KozarezovMsgBuffer, addr KozarezovMsgSuccess,

KozarezovArrayA[esi\*4], KozarezovArrayB[esi\*4], KozarezovArrayC[esi\*4],

KozarezovArrayB[esi\*4], KozarezovArrayC[esi\*4], KozarezovArrayA[esi\*4],

KozarezovArrayA[esi\*4], KozarezovArrayC[esi\*4], KozarezovResult, eax

invoke MessageBox, NULL, addr KozarezovMsgBuffer, addr KozarezovMsgCaption, MB\_OK

inc esi

jmp KozarezovCalcLoop

KozarezovDivZero:

invoke wsprintf, addr KozarezovMsgBuffer, addr KozarezovMsgFail,

KozarezovArrayA[esi\*4], KozarezovArrayB[esi\*4], KozarezovArrayC[esi\*4]

invoke MessageBox, NULL, addr KozarezovMsgBuffer, addr KozarezovMsgCaption, MB\_ICONERROR

inc esi

jmp KozarezovCalcLoop

KozarezovExit:

invoke ExitProcess, 0

end start

**Порівняння розрахунків**

|  |  |  |  |
| --- | --- | --- | --- |
| Формула: | (b/c - 24 + a)/(2\*a\*c - 12) | | |
| Значення змінних | Результат обчислень вручну | Результат обчислень програми | Чи співпало значення? |
| a = 6; b = 60; c = 2; | 1 | 1 | Так |
| a = 4; b = 6; c = 2 | -2 | -2 | Так |
| a = 4; b = 40; c =1 | -5 | -5 | Так |
| a = 2; b = -36; c = 2 | 10 | 10 | Так |
| a = 2; b = 9; c = 3 | ПОМИЛКА | ПОМИЛКА | Так |

**Висновки**

Після виконання лабораторної роботи я не лише ознайомився з основними арифметичними та логічними командами асемблера, а й набув практичного досвіду роботи з масивами даних, організацією циклів та обробкою виняткових ситуацій. Написана програма наочно продемонструвала, як ефективно використовувати регістри процесора для виконання складних обчислень за заданою формулою, а також реалізувати логіку модифікації результатів залежно від їх парності. Особливу увагу було приділено перевірці коректності вхідних даних, що дозволило уникнути помилок, пов'язаних із діленням на нуль. Контрольні розрахунки, проведені для різних комбінацій вхідних параметрів, повністю підтвердили правильність роботи програми, що свідчить про точність реалізованого алгоритму та успішне засвоєння принципів програмування на рівні машинних команд. Ця робота значно розширила моє розуміння низькорівневих обчислень та підтвердила важливість уважного тестування програмного коду.