**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.»

Варіант 21

Виконав: Перевірив:

студент групи ІМ-21 доцент Павлов В.Г.

Сірик Максим Олександрович

Номер у списку групи: 21

**Київ 2024**

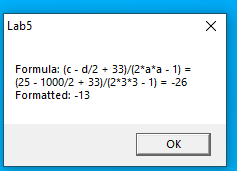
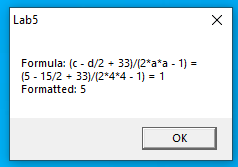
# Визначення варіанту

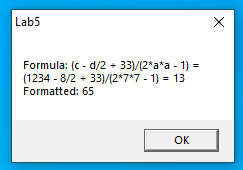
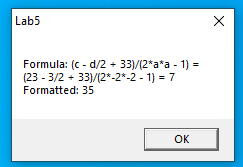
21 % 20 = 1

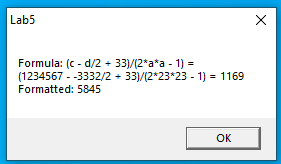
Відповідно вормула для розрахенків:

(c - d/2 + 33)/(2\*a\*a-l);

# Скриншоти







# Розрахунки

1.

**A = 4**

**С = 5**

**D = 15**

(5 – 15 / 2 + 33)/(2\*4\*4-1) = 1

1. 15 / 2 = 7.5 => 7
2. 5 – 7 = -2
3. 33 – 2 = 31
4. 2 \* 4 \* 4 = 32
5. 32 – 1 = 31
6. 31 / 31 = 1

Тоді 1 % 2 = 1 => 1 \* 5 = 5 – formatted value

2.

**A = 3**

**С = 25**

**D = 1000**

(25 - 1000/2 + 33)/(2\*3\*3-l)= -26

1. 1000 / 2 = 500
2. 25 – 500 = -475
3. -475+ 33 = -442
4. 2 \* 3 \* 3 = 18
5. 18 –1 = 17
6. -442 / 17 = -26

Тоді -26 % 2 = 0 => -26 / 2 = -13 – formatted value

3.

**A = -2**

**С = 23**

**D = 3**

(23 - 3/2 + 33)/(2\*(-2)\*(-2)-l)= 7

1. 3 / 2 = 1.5 => 1
2. 23 – 1 = 22
3. 22 + 33 = 55
4. 2 \* (-2) \* (-2) = 8
5. 8 –1 = 7
6. 55 / 7 = 7.857 => 7

Тоді 7 % 2 = 1 => 7 \* 5 = 35 – formatted value

4.

**A = 7**

**С = 1234**

**D = 8**

(1234 - 8/2 + 33)/(2\*7\*7-l)= 7

1. 8 / 2 = 4
2. 1234 – 4 = 1230
3. 1230 + 33 = 1263
4. 2 \* 7 \* 7 = 98
5. 98 –1 = 97
6. 1263 / 97 = 13.02 => 13

Тоді 13 % 2 = 1 => 13 \* 5 = 65 – formatted value

5.

**A = 23**

**С = 1234567**

**D = -3332**

(1234567 - -3332/2 + 33)/(2\* 23 \* 23 -l)= 1169

1. -3332 / 2 = -1666
2. 1234567 – -1666 = 1236233
3. 1230 + 33 = 1236266
4. 2 \* 23 \* 23 = 1058
5. 1058 –1 = 1057
6. 1236266 / 1057 = 1169.5988 => 1169

Тоді 1169 % 2 = 1 => 1169 \* 5 = 5845 – formatted value

# Роздуми

Отже, для даного прикладу неможливо підібрати значення знаменника 0, щоб корені були ірраціональними

2\*a\*a – 1 = 0

2 \* a \* a= 1

A \* a = ½

A = +- sqrt(½)

Що суперечить умові завдання,тому ділення на 0 у моєму варіанті забезпечити не вийде

Також, щоб забезпечити знаменник відємним значенням, a = [-sqrt(½); sqrt(½)] тому забезпечити всі варіанти чисельника і знаменника не використовуючи заборонені значення змінних також не вийде.

# Лістинг програми

Мені дуже шкода, але моя програма не може витримати такого напливе тексту і просто зависає, якщо я вставлю лістинг програми в документ, тому я надішю повну його частину поштою

00000000 .data

00000000 4C 61 62 35 00 MTitle db "Lab5", 0

00000005 46 6F 72 6D 75 my\_formula\_fmt db "Formula: (c - d/2 + 33)/(2\*a\*a - 1) = ", 10, 13,

6C 61 3A 20

28 63 20 2D

20 64 2F 32

20 2B 20 33

33 29 2F 28

32 2A 61 2A

61 20 2D 20

31 29 20 3D

20 0A 0D 28

25 64 20 2D

20 25 64 2F

32 20 2B 20

33 33 29 2F

28 32 2A 25

64 2A 25 64

20 2D 20 31

29 20 3D 20

25 64 0A 0D

46 6F 72 6D

61 74 74 65

64 3A 20 25

64 00

"(%d - %d/2 + 33)/(2\*%d\*%d - 1) = %d", 10, 13,

"Formatted: %d", 0

00000060 00000004 my\_a dd 4, 3, -2, 7, 23

00000003

FFFFFFFE

00000007

00000017

00000074 00000005 my\_c dd 5, 25, 23, 1234, 1234567

00000019

00000017

000004D2

0012D687

00000088 0000000F my\_d dd 15, 1000, 3, 8, -3332

000003E8

00000003

00000008

FFFFF2FC

0000009C = 00000005 array\_length equ ($ - my\_d) / 4 ; equates (cur location start pointer - my\_a's start pointer) / size of array el

0000009C 00000100 [ buff db 256 dup(?)

00

]

0000019C 00000000 counter dd 0

00000000 .code

00000000 my\_calc PROC par\_a:DWORD, par\_c:DWORD, par\_d:DWORD

LOCAL res:DWORD

00000006 8B 5D 08 mov ebx, par\_a ; ebx is a denominator

00000009 0F AF DB imul ebx, ebx ; a\*a

0000000C 6B DB 02 imul ebx, 2 ; 2\*a\*a

0000000F 83 EB 01 sub ebx, 1 ; 2\*a\*a - 1

.if ebx == 0

invoke MessageBox, 0, addr buff, addr MTitle, 0

ret

.endif

0000002D 8B 45 10 mov eax, par\_d ; eax is a numerator

00000030 99 cdq ; sign extend eax into edx:eax

00000031 B9 00000002 mov ecx, 2

00000036 F7 F9 idiv ecx ; d/2

00000038 F7 D8 neg eax ; -d/2

0000003A 03 45 0C add eax, par\_c ; c - d/2

0000003D 83 C0 21 add eax, 33 ; c - d/2 + 33

; eax is ready

00000040 99 cdq

00000041 8B CB mov ecx, ebx

00000043 F7 F9 idiv ecx ; (c - d/2 + 33)/(2\*a\*a - 1)

00000045 89 45 FC mov res, eax

00000048 83 E0 01 and eax, 1

.if eax == 0

0000004F 8B 45 FC mov eax, res

00000052 99 cdq

00000053 B9 00000002 mov ecx, 2

00000058 F7 F9 idiv ecx ; res / 2

.else

0000005C 8B 45 FC mov eax, res

0000005F 6B C0 05 imul eax, 5 ; res \* 5

.endif

invoke wsprintf, ADDR buff, addr my\_formula\_fmt, par\_c, par\_d, par\_a, par\_a, res, eax

invoke MessageBox, 0, addr buff, addr MTitle, 0

ret

0000009B my\_calc ENDP

0000009B main:

.while counter < array\_length

0000009D A1 0000019C R mov eax, counter

invoke my\_calc, my\_a[eax \* 4], my\_c[eax \* 4], my\_d[eax \* 4]

000000BC FF 05 0000019C R inc counter

.endw

invoke ExitProcess, 0

end main

# Висновок

Отже, в ході лабораторної роботи № 5 я навчився використовувати масиви та цілочисельні операції. Всі приклади збігаються із моїми розрахунками. Як і очікувалося при наявності остаці, у цілочисельному діленні, остача відкидається. Я перевіряв парність фактично порівнюючи останній біт у числі,тому що якщо він 1 то число непарне і якщо 0 то парне. Також у мене за варіантом неможливо ділити на 0, і отримати чисельник та знаменник у всіх варіаціях (причина описанна в секції роздуми).