**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 доц. Павлов В.Г.

Групи ІМ-13

Ситник Денис Олександрович

Номер у списку групи: 22

Київ – 2023

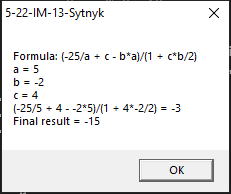
**Мета роботи:** вивчення арифметичних і логічних команд Асемблера і здобуття навичок виконання розрахунків з елементами масивів.

Номер у списку – 22, варіант 2: (-25/a + c - b\*a)/(l + c\*b/2);

**Контрольні розрахунки та скріншоти виконання програми:**

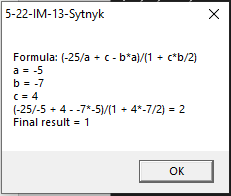
**1) a = 5, b = -2, c = 4**

(-25/a + c - b\*a)/(l + c\*b/2) = (-25/5 + 4 – (-2) \* 5) / (1 + 4 \* (-2) / 2) = (-5 + 14) / -3 = 9 / -3 = -3 – непарне, отже кінцевий результат = -15.



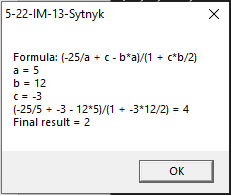
**2) a = -5, b = -7, c = 4**

(-25/a + c - b\*a)/(l + c\*b/2) = (-25/(-5) + 4 – (-7) \* (-5)) / (1 + 4 \* (-7) / 2) = (5 - 31) / (-13) = -26 / -13 = 2 – парне, отже кінцевий результат = 1.



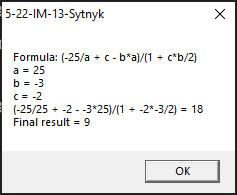
**3) a = 5, b = 12, c = -3**

(-25/a + c - b\*a)/(l + c\*b/2) = (-25/5 – 3 – 12 \* 5) / (1 + (-3) \* 12 / 2) = (-5 -63) / -17 = -68 / -17 = 4 – парне, отже кінцевий результат = 2.



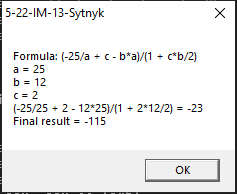
**4) a = 25, b = -3, c = -2**

(-25/a + c - b\*a)/(l + c\*b/2) = (-25/25 – 2 – (-3) \* 25) / (1 + (-2) \* (-3) / 2) = (-1 + 73) / 4 = 72 / 4 = 18 – парне, отже кінцевий результат = 9.



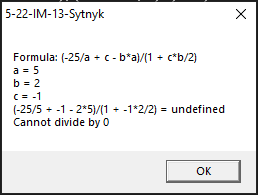
**5) a = 25, b = 12, c = 2**

(-25/a + c - b\*a)/(l + c\*b/2) = (-25/25 + 2 – 12 \* 25) / (1 + 2 \* 12 / 2) = (-1 -298) / 13 = -299 / 13 = -23 – непарне, отже кінцевий результат = -115.



**6) a = 5, b = 2, c = -1**

(-25/a + c - b\*a)/(l + c\*b/2) = (-25/5 + 2 – 2 \* 5) / (1 + (-1) \* 2 / 2) = (-5 -8) / 0 = undefined (ділимо на 0)



**Файл лістингу програми:**

printMessage macro message, title

invoke MessageBox, 0, offset message, offset title, 0

endm

divide macro value, divider

mov eax, value

mov ebx, divider

cdq

idiv ebx

endm

00000000 .data?

00000000 00000001 [ denominator dd 1 dup(?)

00000000

]

00000004 00000040 [ buff db 64 dup(?)

00

]

00000044 00000005 [ result dd 5 dup(?)

00000000

]

00000000 .data

00000000 35 2D 32 32 2D data\_title db "5-22-IM-13-Sytnyk", 0

49 4D 2D 31

33 2D 53 79

74 6E 79 6B

00

00000012 46 6F 72 6D 75 data db "Formula: (-25/a + c - b\*a)/(1 + c\*b/2)", 10,

6C 61 3A 20

28 2D 32 35

2F 61 20 2B

20 63 20 2D

20 62 2A 61

29 2F 28 31

20 2B 20 63

2A 62 2F 32

29 0A 61 20

3D 20 25 69

0A 62 20 3D

20 25 69 0A

63 20 3D 20

25 69 0A 28

2D 32 35 2F

25 69 20 2B

20 25 69 20

2D 20 25 69

2A 25 69 29

2F 28 31 20

2B 20 25 69

2A 25 69 2F

32 29 20 3D

20 25 69 0A

46 69 6E 61

6C 20 72 65

73 75 6C 74

20 3D 20 25

69 0A 00

"a = %i", 10,

"b = %i", 10,

"c = %i", 10,

"(-25/%i + %i - %i\*%i)/(1 + %i\*%i/2) = %i", 10,

"Final result = %i", 10, 0

0000008A 46 6F 72 6D 75 data\_error db "Formula: (-25/a + c - b\*a)/(1 + c\*b/2)", 10,

6C 61 3A 20

28 2D 32 35

2F 61 20 2B

20 63 20 2D

20 62 2A 61

29 2F 28 31

20 2B 20 63

2A 62 2F 32

29 0A 61 20

3D 20 25 69

0A 62 20 3D

20 25 69 0A

63 20 3D 20

25 69 0A 28

2D 32 35 2F

25 69 20 2B

20 25 69 20

2D 20 25 69

2A 25 69 29

2F 28 31 20

2B 20 25 69

2A 25 69 2F

32 29 20 3D

20 75 6E 64

65 66 69 6E

65 64 0A 43

61 6E 6E 6F

74 20 64 69

76 69 64 65

20 62 79 20

30 0A 00

"a = %i", 10,

"b = %i", 10,

"c = %i", 10,

"(-25/%i + %i - %i\*%i)/(1 + %i\*%i/2) = undefined", 10,

"Cannot divide by 0", 10, 0

0000010A 00000005 a\_values dd 5, -5, 5, 5, 5, 5

FFFFFFFB

00000005

00000005

00000005

00000005

00000122 00000004 b\_values dd 4, -7, 12, 10, 12, 2

FFFFFFF9

0000000C

0000000A

0000000C

00000002

0000013A FFFFFFFE c\_values dd -2, 4, -3, -1, -4, -1

00000004

FFFFFFFD

FFFFFFFF

FFFFFFFC

FFFFFFFF

00000000 .code

00000000 main:

00000000 BE 00000000 mov esi, 0

.while esi < 6

;; Calculating denominator

0000000A 8B 04 B5 mov eax, b\_values[esi \* 4]

00000122 R

00000011 8B 0C B5 mov ecx, c\_values[esi \* 4]

0000013A R

00000018 0F AF C1 imul eax, ecx ;; (c\*b)

divide eax, 2 ;; (c\*b/2)

0000001B 8B C0 1 mov eax, eax

0000001D BB 00000002 1 mov ebx, 2

00000022 99 1 cdq

00000023 F7 FB 1 idiv ebx

00000025 83 C0 01 add eax, 1 ;; (1 + c\*b/2)

00000028 A3 00000000 R mov denominator, eax

;; If denominator = 0 show error message

.if denominator == 0

invoke wsprintf, addr buff, addr data\_error,

a\_values[esi \* 4], b\_values[esi \* 4], c\_values[esi \* 4],

a\_values[esi \* 4], c\_values[esi \* 4], b\_values[esi \* 4], a\_values[esi \* 4], c\_values[esi \* 4], b\_values[esi \* 4]

printMessage buff, data\_title

.else

;; Continue calculating

divide -25, a\_values[esi \* 4] ;; (-25/a)

0000009F B8 FFFFFFE7 1 mov eax, -25

000000A4 8B 1C B5 1 mov ebx, a\_values[esi \* 4]

0000010A R

000000AB 99 1 cdq

000000AC F7 FB 1 idiv ebx

000000AE 8B 0C B5 mov ecx, c\_values[esi \* 4]

0000013A R

000000B5 03 C1 add eax, ecx ;; (-25/a + c)

000000B7 8B 1C B5 mov ebx, b\_values[esi \* 4]

00000122 R

000000BE 8B 0C B5 mov ecx, a\_values[esi \* 4]

0000010A R

000000C5 0F AF D9 imul ebx, ecx ;; (b\*a)

000000C8 2B C3 sub eax, ebx ;; (-25/a + c - b\*c)

divide eax, denominator ;; (-25/a + c - b\*a) / (1 + c\*b/2)

000000CA 8B C0 1 mov eax, eax

000000CC 8B 1D 00000000 R 1 mov ebx, denominator

000000D2 99 1 cdq

000000D3 F7 FB 1 idiv ebx

000000D5 A3 00000044 R mov result, eax

;; Checking for result to be even or odd

000000DA A9 00000001 test eax, 1

000000DF 75 02 jnz odd\_num

000000E1 74 05 jz even\_num

;; Processing different types of answers

000000E3 odd\_num:

000000E3 6B C0 05 imul eax, 5 ;; multiply result by 5

000000E6 EB 0A jmp show\_data

000000E8 even\_num:

000000E8 B9 00000002 mov ecx, 2

000000ED 99 cdq

000000EE F7 F9 idiv ecx ;; divide result by 2

000000F0 EB 00 jmp show\_data

000000F2 show\_data:

invoke wsprintf, addr buff, addr data,

a\_values[esi \* 4], b\_values[esi \* 4], c\_values[esi \* 4],

a\_values[esi \* 4], c\_values[esi \* 4], b\_values[esi \* 4], a\_values[esi \* 4], c\_values[esi \* 4], b\_values[esi \* 4],

result, eax

printMessage buff, data\_title

.endif

0000015D 46 inc esi

.endw

end main

**Висновок:**

Виконавши лабораторну роботу з вивчення арифметичних і логічних операцій з цілими числами та масивів, я написав просту віконну програму, яка підраховує результат за заданою формулою та початковими даними.   
Порівнявши результат обчислень програми та попередніх ручних контрольних розрахунків, можна сказати, що програма працює коректно.

Отже, я навчився описувати базові арифметичні і логічні операції мовою асемблер.