

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Кафедра ЕОМ



## **Звіт**

З лабораторної роботи №4

З дисципліни: «Засоби системного програмування»

На тему: «Дослідження роботи команд переходів. Програмування задач з використанням алгоритмів розгалуження.»

Виконав: ст. гр. КІ-201

Добош Маркіян

Прийняв:

Козак Н. Б.

**Львів 2023**

**Мета:** освоїти використання команд порівняння, умовного та безумовного переходів. Набути вміння використовувати арифметичні команди над знаковими даними та команди логічних операцій.

## ЗАВДАННЯ:

1. Створити \*.exe програму, яка реалізовує обчислення, заданого варіантом виразу. Вхідні дані слід вважати цілими числами зі знаком, розміром один байт. Результат обчислення виразу повинен записуватися у пам'ять. Уникнути випадку некоректних обчислень при діленні на нуль та при переповненні розрядної сітки (вивести відповідне текстове повідомлення).
2. За допомогою Debug, відслідкувати правильність виконання програми (продемонструвати результати проміжних та кінцевих обчислень) та проаналізувати отримані результати для різних вхідних даних.
3. Скласти звіт про виконану роботу з приведенням тексту програми та коментарів до неї.

**Мій варіант:**

7	$X = \begin{cases} (4 * b - 1) / a, & a > b, \\ 255, & a = b, \\ (a - 5) / b, & a < b. \end{cases}$
---	---

Лістинг програми:

```
.686
.model flat, stdcall
option casemap:none
include C:\masm32\include\kernel32.inc
include C:\masm32\include\user32.inc
includelib C:\masm32\lib\kernel32.lib
includelib C:\masm32\lib\user32.lib
.data
A db 10
B db -100
X dw 0
Hello db 13, 10, ' X = (4*b-1)/a if a > b', 13, 10
db ' X = 255 if a = b', 13, 10
db ' X = (a-5)/b if a < b', 13, 10
Operands db 13, 10, ' A =      B =      ', 13, 10
NumberOfCharsToWrite_Hello dd $-Hello
Error db 13, 10, 'Error - divide by zero!', 13, 10
NumberOfCharsToWrite_Error dd $-Error
Result db ' X = ', 16 dup(0)
NumberOfCharsToWrite_Result dd $-Result
format db '%hd', 0
hConsoleOutput dd 0
NumberOfCharsWritten dd 0
.code

start:
;вивід повідомлення Hello
mov al, A
cbw
push ax
```

```

push offset format
push offset [Operands+7]
call wsprintfA
mov al, B
cbw
push ax
push offset format
push offset [Operands+15]
call wsprintfA
push -11
call GetStdHandle
mov hConsoleOutput, eax
push 0
push offset NumberOfCharsWritten
push NumberOfCharsToWrite_Hello
push offset Hello
push hConsoleOutput
call WriteConsoleA

```

;перевірка на рівність A і B

```

mov al, A
cmp al, B
jne A_ne_B
mov X, 255
jmp Output_Result

```

```

A_ne_B:
jg A_g_B
cmp B, 0
je Output_Error

```

;обчислення X при a<b

```

cbw
mov bx, ax
mov al, A
cbw
sub ax, 5
cwd
movsx bx, B
idiv bx
mov X, ax
jmp Output_Result

```

;обчислення X при a>b

```

A_g_B:
cmp A, 0
je Output_Error

```

```

mov al, B
mov bx, 4
cbw
imul bx
dec ax

```

```

mov cx, ax
mov al, A
cbw
mov bx, ax
mov ax, cx

```

```

cld
idiv bx
mov X, ax

jmp Output_Result

jmp Output_Result

;вивід результату

Output_Result:
push X
push offset format
push offset [Result+5]
call wsprintfA
push offset NumberOfCharsWritten
push NumberOfCharsToWrite_Result
push offset Result
push hConsoleOutput
call WriteConsoleA
jmp exit

;вивід повідомлення про ділення на нуль

Output_Error:
push offset NumberOfCharsWritten
push NumberOfCharsToWrite_Error
push offset Error
push hConsoleOutput
call WriteConsoleA
jmp exit

;вихід з програми

exit:
push 0
call ExitProcess
end start

```

```

X = (4*b-1)/a if a > b
X = 255 if a = b
X = (a-5)/b if a < b

A = -10 B = 10
X = -1          Press any key to continue . . .

```

*Результат виконання програми, коли  $a < b$*

```

X = (4*b-1)/a if a > b
X = 255 if a = b
X = (a-5)/b if a < b

A = 10 B = 10
X = 255          Press any key to continue . . .

```

*Результат виконання програми, коли  $a == b$*

```
X = (4*b-1)/a if a > b
X = 255 if a = b
X = (a-5)/b if a < b

A = 10  B = -100
X = -40      Press any key to continue . . . ■
```

*Результат виконання програми, коли  $a > b$*

**Висновок:** виконуючи дану лабораторну роботу я отримав знання з використання команд порівняння, умовного та безумовного переходів. Набув вміння використовувати арифметичні команди над знаковими даними та команди логічних операцій.