**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

Арифметичні і логічні операції з цілими числами. Масиви.

з дисципліни

Системне програмування

Виконав: Перевірив:

cтудент групи ІП-03 доц. кафедри ОТ

Янишин Віталій Павлов В.Г

Номер залікової книжки: 0327

Номер у списку групи: 27

Київ 2022

**Мета лабораторної роботи:** Вивчення арифметичних і логічних команд Асемблера і здобуття навиків виконання розрахунків з елементами масивів.

**Хід роботи**

Варіант:

|  |  |
| --- | --- |
| **3** | **(c/d + 3\*a/2)/(c - a + 1);** |

ОДЗ

d!=0;

c-a+1!=0;

Підібрані змінні (a, d, c)

A = { 2, 12, -18, -70, -112 }

C = { 6, 10, 6, -60, -30 }

D = { 3, -2, 3, 12, -15 }

Контрольні розрахунки

( c/1d +4 3\*2a/32 ) /7 ( c-5a+61 )

Цифри після арифметичних знаків символізують порядок виконання дій

Перша підстановка:

( c/d + 3\*a/2 ) / ( c-a+1 ) = ( 6/3 + 3\*2/2 ) / ( 6-2+1)

1. 6/3=2
2. 3\*2=6
3. 6/2=3
4. 2+3=5
5. 6-2=4
6. 4+1=5
7. 5/5=1

Результат: 1

Число непарне, отже множимо на 5 і кінцевий результат:

1\*5=5

Друга підстановка:

( c/d + 3\*a/2 ) / ( c-a+1 ) = ( 10/(-2) + 3\*12/2 ) / ( 10-12+1)

1. 10/(-2)=-5
2. 3\*12=36
3. 36/2=18
4. -5+18=13
5. 10-12=-2
6. -2+1=-1
7. 13/(-1)=-13

Результат: -13

Число непарне, отже множимо на 5 і кінцевий результат:

-13\*5=-65

Третя підстановка:

( c/d + 3\*a/2 ) / ( c-a+1 ) = ( 6/3 + 3\*(-18)/2 ) / ( 6-(-18)+1)

1. 6/3=2
2. 3\*(-18)=-54
3. -54/2=-27
4. 2+(-27)=-25
5. 6-(-18)=24
6. 24+1=25
7. -25/25=-1

Результат: -1

Число непарне, отже множимо на 5 і кінцевий результат:

-1\*5=-5

Четверта підстановка:

( c/d + 3\*a/2 ) / ( c-a+1 ) = ( -60/12 + 3\*(-70)/2 ) / ( -60-(-70)+1)

1. -60/12=-5
2. 3\*(-70)=-210
3. -210/2=-105
4. -5+(-105)=-110
5. -60-(-70)=10
6. 10+1=11
7. -110/11=-10

Результат: -10

Число парне, отже ділимо на 2 і кінцевий результат:

-10/2=-5

П’ята підстановка:

( c/d + 3\*a/2 ) / ( c-a+1 ) = ( -30/(-15) + 3\*(-112)/2 ) / ( -30-(-112)+1)

1. -30/(-15)=2
2. 3\*(-112)=-336
3. -336/2=-168
4. 2+(-168)=-166
5. -30-(-112)=82
6. 82+1=83
7. -166/83=-2

Результат: -2

Число парне, отже ділимо на 2 і кінцевий результат:

-2/2=-1

Підбиваємо підсумки підрахунків:

Result = { 1, -13, -1, -10, -2 }

EndResult = { 5, -65, -5, -5, -1 }

Лістинг програми на мові Ассемблера

.386

.model flat, stdcall

option casemap :none

include \masm32\include\masm32rt.inc

.data?

Buff db 256 dup(?)

GlobalBuff db 512 dup(?)

.data

;arrays of variables

A\_values dd 2,12, -18,-70,-112

C\_values dd 6,10, 6,-60,-30

D\_values dd 3,-2, 3,12,-15

MsgBoxCaption db "1-23-IP-03-Yanyshyn", 0

MyVariant db "Variant 3: (c/d+3\*a/2)/(c-a+1).", 10,10, 0

PreviewAndResult db "(%d / %d + 3 \* %d / 2)/(%d - %d + 1) = %d/%d", 10,

"Result: %d", 10, "EndResult: %d", 10, 10, 0

Error db "Change variables", 10, 10, 0

Denominator dd 0

Numerator dd 0

Results dd 5 dup(?)

EndResult dd 5 dup(?)

.code

Main:

mov edi, 0; counter

invoke wsprintf, offset GlobalBuff, offset MyVariant

.while edi < 5

mov ecx, C\_values[4 \* edi]

mov ebx, A\_values[4 \* edi]

sub ecx, ebx ;c-a -> ecx

inc ecx ;c-a+1 -> ecx

mov Denominator, ecx

mov eax, C\_values[4 \* edi]

mov ebx, D\_values[4 \* edi]

.if ebx!=0 && Denominator!=0

cdq

idiv ebx ; c/d -> eax

mov Numerator, eax

mov ecx, A\_values[4 \* edi]

mov eax, 3

cdq

imul eax, ecx ; 3\*a -> eax

mov ebx, 2

cdq

idiv ebx ;3\*a/2 -> eax

add Numerator, eax

mov eax, Numerator

mov ecx, Denominator

cdq

idiv ecx ; result -> eax

mov Results[4\*edi], eax

test eax, 1

jnz oddResult

jz evenResult

evenResult:

mov ebx, 2

cdq

idiv ebx

mov EndResult[4\*edi], eax

jmp fillBuf

oddResult:

imul eax, 5

mov EndResult[4\*edi], eax

jmp fillBuf

fillBuf:

invoke wsprintf, offset Buff, offset PreviewAndResult,

C\_values[4 \* edi], D\_values[4 \* edi],

A\_values[4 \* edi], C\_values[4 \* edi],

A\_values[4 \* edi], Numerator, Denominator,

Results[4 \* edi], EndResult[4 \* edi]

.else

invoke wsprintf, offset Buff, offset Error

.endif

invoke szCatStr, offset GlobalBuff, offset Buff

inc edi

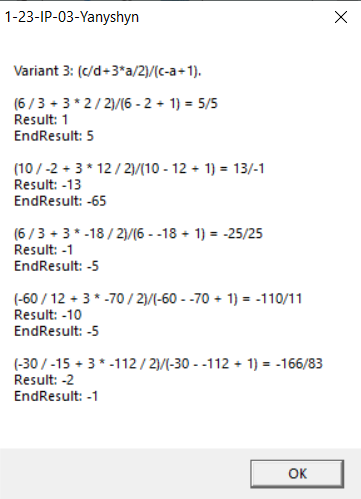
.endw

invoke MessageBox, 0, addr GlobalBuff, addr MsgBoxCaption, 0

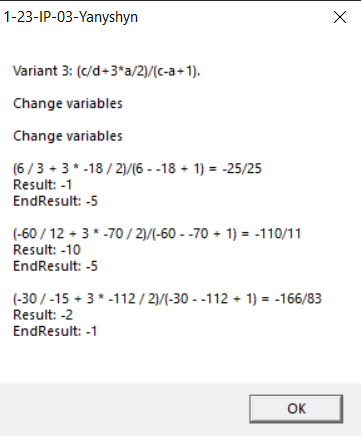
end Main

Результати тестування

Підставлення підібраних змінних



Спроба підставити 0 на місце D в першому прикладі та 3 і 4 на місце с і а відповідно у другому прикладі



Висновок

На лабораторній роботі я вивчив арифметичні і логічні команд Асемблера і здобув навиків виконання розрахунків з елементами масивів. Оскільки ми працювали з цілими числами без округлень результати контрольних розрахунків та результат програми ідентичні