

ЛАБОРАТОРНА РОБОТА № 5

ОБЧИСЛЕННЯ СКЛАДНИХ ЦІЛОЧИСЛЕНИХ ВИРАЗІВ НА МОВІ ASSEMBLER

Мета заняття: ознайомитися з основними командами мови Assembler для обчислення складних цілочисельних виразів; набути практичних навичок в написанні програм для обчислення складних цілочисельних виразів на мові Assembler.

Хід роботи:

1. Написати програму для обчислення заданого цілочисленого виразу(табл. 5.5) для початкових даних в знаковому форматі довжиною 8 біт, використовуючи арифметичні операції ADD, ADC, INC, SUB, SBB, DEC, NEG, IMUL, IDIV, CBW, CWD. Провести тестові перевірки, відмітити нормальні та аномальні результати. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні

Лістинг програми:

```
#include "stdafx.h"
#include <stdio.h>
#include <stdlib.h>

int main()
{
    signed char a, f, b, c, e, res_c, res_asm;
    printf("a = "); scanf_s("%d", &a);
    printf("b = "); scanf_s("%d", &b);
    printf("c = "); scanf_s("%d", &c);
    printf("e = "); scanf_s("%d", &e);
    printf("f = "); scanf_s("%d", &f);
    res_c = (b + c * b - a / 4 - e) / (a * b - 1 + f);
    printf("Result C = %d\n", res_c);
    __asm {
        mov al, b;
        mov bl, c;
        imul bl;
        mov bl, 1;
        idiv bl; //al=c*b
        add al, b;
        mov cl, al;
        mov al, a;
        cbw;
        mov bl, 4;
        idiv bl; //al=a/4
        sub cl, bl;
        sub cl, e;
        mov al, a;
        mov bl, b;
```

					ЖДТУ.18.125.15.000 – Лр5		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.	Макрарчук В.В				Звіт з лабораторної роботи №5	Літ.	Арк.
Перевір.	Байлюк Є.М						1
Керівник						Аркушів	
Н. контр.						8	
Зав. каф.						ФІКТ Гр. КБ-2[1]	

```

    imul bl;
    mov bl, 1;
    idiv bl;
    dec al;
    add al, f;
    xchg al, cl;
    cbw;
    idiv cl;
    mov res_asm, al;
}
printf("Result ASM = %d\n", res_asm);
system("Pause");
return 0;
}Результат виконання програми:

```

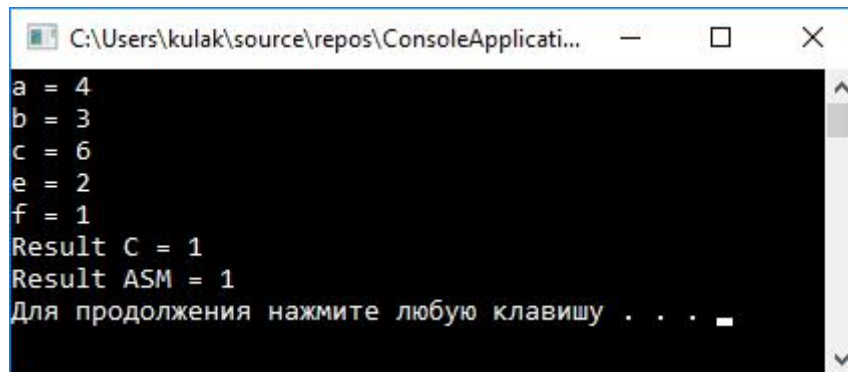


Рис. 1. Результат виконання програми

Значення регістрів при покроковому виконанні

Крок	Команда	Значення регістру			EFLAGS/FLAGS (CF, OF)
		al	cl	bl	
1	mov al, b	3	н/в	н/в	—
2	mov bl, c	3	н/в	6	—
3	imul bl	18	н/в	н/в	—
4	mov bl, 1	18	н/в	1	—
5	idiv bl	18	н/в	1	—
6	add al, b	21	н/в	1	—
7	mov cl, al	21	21	1	—
8	mov al, a	4	21	1	—
9	mov bl, 4	4	21	4	—

10	idiv bl	1	21	4	—
11	sub cl, al	1	20	4	—
12	sub cl, e	1	20	4	—
13	mov al, a	4	20	4	—
14	mov bl, b	4	20	3	—
15	imul bl	12	20	3	—
16	mov bl, 1	12	20	1	—
17	idiv bl	12	20	1	—
18	dec al	11	20	1	—
19	add al, f	12	20	1	—
20	xchg al, cl	20	12	1	—
21	idiv cl	1	12	1	—
22	mov res_asm, al	1	12	1	

2. Написати програму для обчислення заданого цілочисленого виразу (табл. 5.5) для початкових даних в знаковому форматі довжиною 16 біт, використовуючи арифметичні операції ADD, ADC, INC, SUB, SBB, DEC, NEG, IMUL, IDIV, CBW, CWD. Провести тестові перевірки, відмітити нормальні та аномальні результати. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

Лістинг програми:

```
#include "pch.h"
#include <stdio.h>
#include <stdlib.h>

int main()
{
    short int a, f, b, c, e, res_c, res_asm;
    printf("a = "); scanf_s("%d", &a);
    printf("b = "); scanf_s("%d", &b);
    printf("c = "); scanf_s("%d", &c);
    printf("e = "); scanf_s("%d", &e);
    printf("f = "); scanf_s("%d", &f);
    res_c = (b + c * b - a / 4 - e) / (a * b - 1 + f);
```

```

printf("Result C = %d\n", res_c);
__asm {
    mov ax, b;
    mov bx, c;
    imul bx;
    mov bx, 1;
    idiv bx; //ax=c*b
    add ax, b;
    mov cx, ax;
    mov ax, a;
    mov bx, 4;
    idiv bx; //ax=a/4
    sub cx, ax;
    sub cx, e;
    mov ax, a;
    mov bx, b;
    imul bx;
    mov bx, 1;
    idiv bx;
    dec ax;
    add ax, f;
    xchg ax, cx;
    idiv cx;
    mov res_asm, ax;
}
printf("Result ASM = %d\n", res_asm);
system("Pause");
return 0;

```

Результат виконання програми:

Рис. 2. Результат виконання програми

Значення регістрів при покроковому виконанні

Крок	Команда	Значення регістру			EFLAGS/FLAGS (CF, OF)
		ax	cx	bx	
1	mov ax, b	30	н/в	н/в	—
2	mov bx, c	30	н/в	60	—
3	imul bx	180	н/в	н/в	—

4	mov bx, 1	180	н/в	10	—
5	idiv bx	180	н/в	10	—
6	add ax, b	210	н/в	10	—
7	mov cx, ax	210	210	10	—
8	mov ax, a	40	210	10	—
9	mov bx, 4	40	210	40	—
10	idiv bx	10	210	40	—
11	sub cx, ax	10	200	40	—
12	sub cx, e	10	200	40	—
13	mov ax, a	40	200	40	—
14	mov bx, b	40	200	30	—
15	imul bx	120	200	30	—
16	mov bx, 1	120	200	10	—
17	idiv bx	120	200	10	—
18	dec ax	110	200	10	—
19	add ax, f	120	200	10	—
20	xchg ax, cx	200	120	10	—
21	idiv cx	1	120	10	—
22	mov res_asm, ax	1	120	10	

3. Написати програму для обчислення заданого цілочисленого виразу(табл. 5.5) для початкових даних в знаковому форматі довжиною 32 біт, використовуючи арифметичні операції ADD, ADC, INC, SUB, 12 SBB, DEC, NEG, IMUL, IDIV, CBW, CWD. Провести тестові перевірки, відмітити нормальні та аномальні результати. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

Лістинг програми:

```
#include "pch.h"
#include <stdio.h>
#include <cstdlib>

int main()
{
    int a, f, b, c, e, res_c, res_asm;
    printf("a = "); scanf_s("%d", &a);
    printf("b = "); scanf_s("%d", &b);
    printf("c = "); scanf_s("%d", &c);
    printf("e = "); scanf_s("%d", &e);
    printf("f = "); scanf_s("%d", &f);
    res_c = (b + c * b - a / 4 - e) / (a * b - 1 + f);
    printf("Result C = %d\n", res_c);
    __asm {
        mov eax, b;
        mov ebx, c;
        imul ebx;
        mov ebx, 1;
        idiv ebx; //ax=c*b
        add eax, b;
        mov ecx, eax;
        mov eax, a;
        mov ebx, 4;
        idiv ebx; //ax=a/4
        sub ecx, eax;
        sub ecx, e;
        mov eax, a;
        mov ebx, b;
        imul ebx;
        mov ebx, 1;
        idiv ebx;
        dec eax;
        add eax, f;
        xchg eax, ecx;
        idiv ecx;
        mov res_asm, eax;
    }
    printf("Result ASM = %d\n", res_asm);
    system("Pause");
    return 0;
}
```

Результат виконання програми:

```
C:\Users\kulak\source\repos\ConsoleApplication35\Debu...
a = 40000
b = 30000
c = 60000
e = 20000
f = 10000
Result C = 1
Result ASM = 1
Для продолжения нажмите любую клавишу . . .
```

Рис. 3. Результат виконання програми

Значення регістрів при покроковому виконанні

Крок	Команда	Значення регістру			EFLAGS/FLAGS (CF, OF)
		eax	ecx	ebx	
1	mov eax, b	30000	н/в	н/в	—
2	mov ebx, c	30000	н/в	60000	—
3	imul ebx	180000	н/в	н/в	—
4	mov ebx, 1	180000	н/в	10000	—
5	idiv ebx	180000	н/в	10000	—
6	add eax, b	210000	н/в	10000	—
7	mov ecx, eax	210000	210000	10000	—
8	mov eax, a	40000	210000	10000	—
9	mov ebx, 4	40000	210000	40000	—
10	idiv ebx	10000	210000	40000	—
11	sub ecx, eax	10000	200000	40000	—
12	sub ecx, e	10000	200000	40000	—
13	mov eax, a	40000	200000	40000	—
14	mov ebx,b	40000	200000	30000	—
15	imul ebx	120000	200000	30000	—
16	mov ebx, 1	120000	200000	10000	—
17	idiv ebx	120000	200000	10000	—
18	dec eax	110000	200000	10000	—
19	add eax, f	120000	200000	10000	—
20	xchg eax,ecx	200000	120000	10000	—
21	idiv ecx	1	120000	10000	—
22	mov res_asm, ax	1	120000	10000	

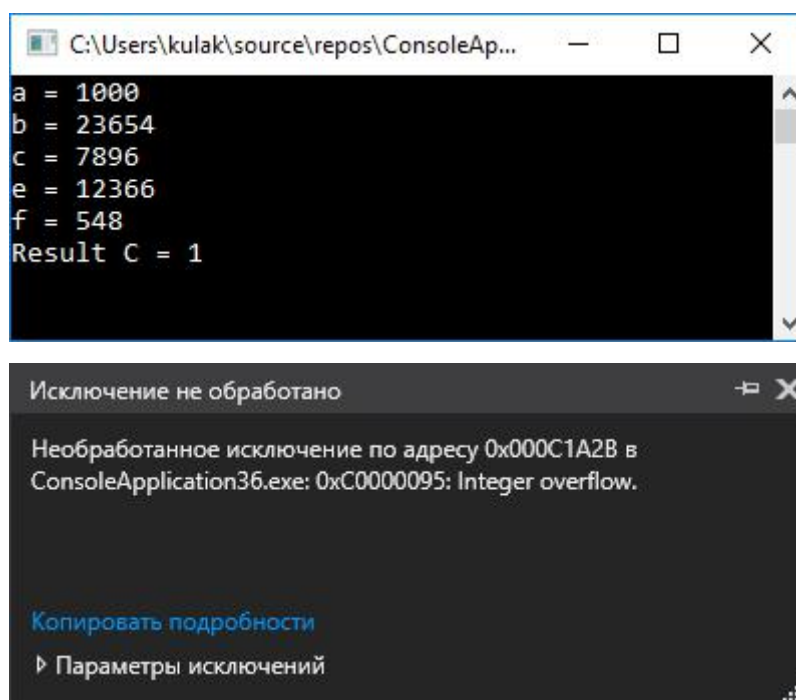


Рис 4. Помилка при введенні чисел поза діапазоном

Висновки: в ході виконання лабораторної роботи було ознайомлено з основними командами мови Assembler для обчислення складних цілочисельних виразів; набули практичних навичок в написанні програм для обчислення складних цілочисельних виразів на мові Assembler

		Макрарчук В.В			ЖДТУ.18.125.15.000 – Лр5	Арк.
		Байлюк Є.М				8
Змн.	Арк.	№ докум.	Підпис	Дата		