

ЛАБОРАТОРНА РОБОТА №4

Обчислення простих цілочисельних виразів на мові Assembler

Мета: ознайомитися з типами цілочисельних даних платформ Win16 та Win32; ознайомитися з основними командами мови Assembler; набути практичних навичок в написанні програм для обчислення простих цілочисельних виразів на мові Assembler.

Хід роботи:

Завдання 1. Написати програму для обчислення заданого цілочисленого виразу для початкових даних в знаковому форматі довжиною 8 біт, використовуючи арифметичні операції ADD, INC, SUB, DEC, NEG. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

15	$-(a+b-1)+(c-d)+e-f+6$
----	------------------------

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <tchar.h>

int main(int argc, _TCHAR* argv[])
{
    char a, b, c, d, e, f, res_c, res_asm;
    printf("a = "); scanf_s("%d", &a);
    printf("b = "); scanf_s("%d", &b);
    printf("c = "); scanf_s("%d", &c);
    printf("d = "); scanf_s("%d", &d);
    printf("e = "); scanf_s("%d", &e);
    printf("f = "); scanf_s("%d", &f);
    res_c = -(a+b-1)+(c-d)+e-f+6;
    printf("Result C = %d\n", res_c);
    __asm {
        mov al, a;      // <al> = a
        add al, b;      // <al> = a + b
        sub al, 1;      // <al> = a + b - 1
        neg al;         // <al> = -(a + b - 1)
        mov cl, c;      // <cl> = c
        sub cl, d;      // <cl> = c-d
        add al, cl;     // <al> = -(a + b - 1) + (c-d)
        add al, e;      // <al> = -(a + b - 1) + (c-d) + e
        sub al, f;      // <al> = -(a + b - 1) + (c-d) + e - f
        add al, 6;      // <al> = -(a + b - 1) + (c-d) + e - f + 6
        mov res_asm, al;
    }
}
```

					ЖДТУ.18.125.15.000 – Лр4				
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №4	Літ.	Арк.	Аркушів	
Розроб.	Макарчук В,В						1	6	
Перевір.	Байлюк Є,М					ФІКТ Гр. КБ-2 [1]			
Керівник									
Н. контр.									
Зав. каф.									

```

printf("Result ASM= %d\n", res_asm);
system("Pause");
return 0;
}

```

Результат виконання програми:

```

a = 13
b = 12
c = 11
d = 10
e = 9
f = 8
Result C = -16
Result ASM= -16
Для продовження натисніть будь-яку клавішу . . .

```

Рис. 1. Результат виконання програми

Значення регістрів при покроковому виконанні:

Крок	Команда	Значення регістра				EFLAGS/FLAGS (CF, OF)
		al	bl	cl	dl	
1	mov al, a	13	н/в	н/в	н/в	---
2	add al, b	25	н/в	н/в	н/в	---
3	sub al, 1	24	н/в	н/в	н/в	---
4	neg al	-24	н/в	н/в	н/в	---
5	mov cl, c	н/в	н/в	11	н/в	---
6	sub cl, d	н/в	н/в	1	н/в	---
7	add al, ecx	-23	н/в	н/в	н/в	---
8	add al, e	-14	н/в	н/в	н/в	---
9	sub al, f	-22	н/в	н/в	н/в	---
10	add al, 6	-16	н/в	н/в	н/в	---
11	mov res_asm, al	-16	н/в	н/в	н/в	---

Завдання 2. Написати програму для обчислення заданого цілочисленого виразу(табл.4.6) для початкових даних в знаковому форматі довжиною 16 біт, використовуючи арифметичні операції ADD, INC, SUB, DEC, NEG. Виконати

покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <tchar.h>

int main(int argc, _TCHAR* argv[])
{
    short int a, b, c, d, e, f, res_c, res_asm;
    printf("a = "); scanf_s("%d", &a);
    printf("b = "); scanf_s("%d", &b);
    printf("c = "); scanf_s("%d", &c);
    printf("d = "); scanf_s("%d", &d);
    printf("e = "); scanf_s("%d", &e);
    printf("f = "); scanf_s("%d", &f);
    res_c = -(a+b-1)+(c-d)+e-f+6;
    printf("Result C = %d\n", res_c);
    __asm {
        mov ax, a;      // <ax> = a
        add ax, b;      // <ax> = a + b
        sub ax, 1;      // <ax> = a + b - 1
        neg ax;         // <ax> = -(a + b - 1)
        mov cx, c;      // <cx> = c
        sub cx, d;      // <cx> = c-d
        add ax, cx;     // <ax> = -(a + b - 1) + (c-d)
        add ax, e;      // <ax> = -(a + b - 1) + (c-d) + e
        sub ax, f;      // <ax> = -(a + b - 1) + (c-d) + e - f
        add ax, 6;      // <ax> = -(a + b - 1) + (c-d) + e - f + 6
        mov res_asm, ax;
    }

    printf("Result ASM= %d\n", res_asm);
    system("Pause");
    return 0;
}
```

Результат виконання програми:

```
a = 13
b = 12
c = 11
d = 10
e = 9
f = 8
Result C = -16
Result ASM= -16
Для продовження натисніть будь-яку клавішу . . .
```

Рис. 2. Результат виконання програми

Значення регістрів при покроковому виконанні:

Крок	Команда	Значення регістра	EFLAGS/FLAGS
------	---------	-------------------	--------------

		Макаруч В.В			ЖДТУ.18.125.15.000 – Лр4	Арк.
		Байлюк С.М				
Змн.	Арк.	№ докум.	Підпис	Дата		3

		ax	cx	(CF, OF)
1	mov ax, a	13	Н/В	---
2	add ax, b	25	Н/В	---
3	sub ax, 1	24	Н/В	---
4	neg ax	-24	Н/В	---
5	mov cx, c	Н/В	11	---
6	sub cx, d	Н/В	1	---
7	add ax, ecx	-23	Н/В	---
8	add ax, e	-14	Н/В	---
9	sub ax, f	-22	Н/В	---
10	add ax, 6	-16	Н/В	---
11	mov res_asm, ax	-16	Н/В	---

Завдання 3. Написати програму для обчислення заданого цілочисленого виразу(табл.4.6) для початкових даних в знаковому форматі довжиною 32 біт, використовуючи арифметичні операції ADD, INC, SUB, DEC, NEG. Виконати покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <tchar.h>

int main(int argc, _TCHAR* argv[])
{
    int a, b, c, d, e, f, res_c, res_asm;
    printf("a = "); scanf_s("%d", &a);
    printf("b = "); scanf_s("%d", &b);
    printf("c = "); scanf_s("%d", &c);
    printf("d = "); scanf_s("%d", &d);
    printf("e = "); scanf_s("%d", &e);
    printf("f = "); scanf_s("%d", &f);
    res_c = -(a+b-1)+(c-d)+e-f+6;
    printf("Result C = %d\n", res_c);
    __asm {
        mov eax, a;      // <eax> = a
        add eax, b;      // <eax> = a + b
        sub eax, 1;      // <eax> = a + b - 1
        neg eax;         // <eax> = -(a + b - 1)
        mov ecx, c;      // <ecx> = c
        sub ecx, d;      // <ecx> = c-d
        add eax, ecx;    // <eax> = -(a + b - 1) + (c-d)
```

		Макарчук В.В			ЖДТУ.18.125.15.000 – Лр4	Арк.
		Байлюк Є.М				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

    add eax, e; // <eax>=-(a + b - 1)+(c-d) +e
    sub eax, f; // <eax>=-(a + b - 1)+(c-d) +e-f
    add eax, 6; //<eax>=-(a + b - 1)+(c-d) +e-f+6
    mov res_asm, eax;
}

printf("Result ASM= %d\n", res_asm);
system("Pause");
return 0;
}

```

Результат виконання програми:

```

a = 13
b = 12
c = 11
d = 10
e = 9
f = 8
Result C = -16
Result ASM= -16
Для продовження натисніть будь-яку клавішу . . .

```

Рис. 3. Результат виконання програми

Значення регістрів при покроковому виконанні:

Крок	Команда	Значення регістра		EFLAGS/FLAGS (CF, OF)
		eax	ecx	
1	mov eax, a	13	Н/В	---
2	add eax, b	25	Н/В	---
3	sub eax, 1	24	Н/В	---
4	neg eax	-24	Н/В	---
5	mov ecx, c	Н/В	11	---
6	sub ecx, d	Н/В	1	---
7	add eax, ecx	-23	Н/В	---
8	add eax, e	-14	Н/В	---
9	sub eax, f	-22	Н/В	---
10	add eax, 6	-16	Н/В	---
11	mov res_asm, eax	-16	Н/В	---

Висновок: Ми набули навичок роботи з одновимірними масивами.

		Макарчук В.В			ЖДТУ.18.125.15.000 – Лр4	Арк.
		Байлюк Є.М				
Змн.	Арк.	№ докум.	Підпис	Дата		6