

## 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.

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

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 - c) - (d + 1) - (e - f);
    printf("Result C = %d\n", res_c);
    __asm {
        mov al, a;      // <al> = a
        add al, b;      // <al> = a + b
        sub al, c;      // <al> = a + b - c
        neg al;         // <al> = -(a + b - c)
        mov cl, d;      // <cl> = d
        inc cl;         // <cl> = d + 1
        sub al, cl;     // <al> = -(a + b - c) - (d + 1)
        mov dl, e;      // <dl> = e
        sub dl, f;      // <dl> = e - f
        sub al, dl;     // <al> = -(a + b - c) - (d + 1) - (e - f)
        mov res_asm, al;
    }
}
```

					ЖДТУ.18.121.01.000 – Лр4			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Абанін К.А				Звіт з лабораторної роботи №4	Літ.	Арк.	Аркушів
Перевір.	Байлюк С,М						1	6
Керівник						ФІКТ Гр. ПІ-60 [1]		
Н. контр.								
Зав. каф.								

```

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

```

Результат виконання програми:

```

a = 11
b = 10
c = 9
d = 8
e = 7
f = 6
Result C = -22
Result ASM= -22
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	11	н/в	н/в	н/в	---
2	add al, b	21	н/в	н/в	н/в	---
3	sub al, c	12	н/в	н/в	н/в	---
4	neg al	-12	н/в	н/в	н/в	---
5	mov cl, d	н/в	н/в	8	н/в	---
6	inc cl	н/в	н/в	9	н/в	---
7	add al, cl	-21	н/в	н/в	н/в	---
8	mov dl, e	н/в	н/в	н/в	7	---
9	sub dl, f	1	н/в	н/в	н/в	---
10	sub al, dl	-22	н/в	н/в	н/в	---
11	mov res_asm, al	-22	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

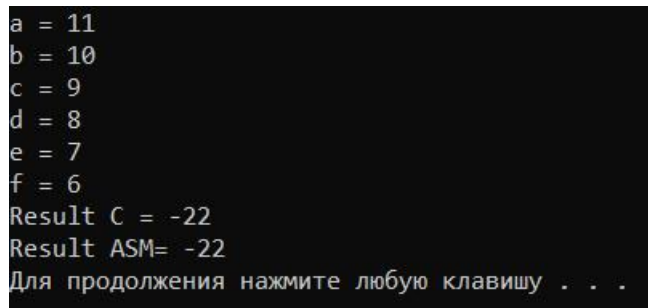
покрокове виконання асемблерного коду та навести значення регістрів при їх виконанні.

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 - c) - (d + 1) - (e - f);
    printf("Result C = %d\n", res_c);
    __asm {
        mov ax, a; // <ax> = a
        add ax, b; // <ax> = a + b
        sub ax, c; // <ax> = a + b - c
        neg ax; // <ax> = -(a + b - c)
        mov cx, d; // <cx> = d
        inc cx; // <cx> = d + 1
        sub ax, cx; // <ax> = -(a + b - c) - (d + 1)
        mov dx, e; // <dx> = e
        sub dx, f; // <dx> = e - f
        sub ax, dx; // <ax> = -(a + b - c) - (d + 1) - (e - f)
        mov res_asm, ax;
    }
    printf("Result ASM= %d\n", res_asm);
    system("Pause");
    return 0;
}
```

Результат виконання програми:



```
a = 11
b = 10
c = 9
d = 8
e = 7
f = 6
Result C = -22
Result ASM= -22
Для продолжения нажмите любую клавишу . . .
```

Рис. 2. Результат виконання програми

Значення регістрів при покроковому виконанні:

Крок	Команда	Значення регістра				EFLAGS/FLAGS (CF, OF)
		ax	bx	cx	dx	
1	mov ax, a	11	н/в	н/в	н/в	---
2	add ax, b	21	н/в	н/в	н/в	---
3	sub ax, c	12	н/в	н/в	н/в	---
4	neg ax	-12	н/в	н/в	н/в	---
5	mov cx, d	н/в	н/в	8	н/в	---
6	Inc cx	н/в	н/в	9	н/в	---
7	add ax, cx	-21	н/в	н/в	н/в	---
8	mov dx, e	н/в	н/в	н/в	7	---
9	sub dx, f	1	н/в	н/в	н/в	---
10	sub ax, dx	-22	н/в	н/в	н/в	---
11	mov res_asm, ax	-22	н/в	н/в	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 - c) - (d + 1) - (e - f);
    printf("Result C = %d\n", res_c);
    __asm {
        mov eax, a;        // <eax> = a
        add eax, b;        // <eax> = a + b
    }
```

```

sub eax, c;      // <eax> = a + b - c
neg eax;        // <eax> = -(a + b - c)
mov ecx, d;     //<ecx>= d
inc ecx;        //<ecx>=d+1
sub eax, ecx;   // <eax>=-(a + b - c) -(b+1)
mov edx, e;     //<edx>=e
sub edx, f;     //<edx>=e-f
sub eax, edx;   //<eax>= -(a + b - c) - (d + 1) - (e - f)
mov res_asm, eax;
}
printf("Result ASM= %d\n", res_asm);
system("Pause");
return 0;
}

```

Результат виконання програми:

```

a = 11
b = 10
c = 9
d = 8
e = 7
f = 6
Result C = -22
Result ASM= -22
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	ebx	ecx	edx	
1	mov eax, a	11	н/в	н/в	н/в	---
2	add eax, b	21	н/в	н/в	н/в	---
3	sub eax, c	12	н/в	н/в	н/в	---
4	neg eax	-12	н/в	н/в	н/в	---
5	mov ecx, d	н/в	н/в	8	н/в	---
6	inc ecx	н/в	н/в	9	н/в	---
7	add eax, ecx	-21	н/в	н/в	н/в	---
8	mov edx, e	н/в	н/в	н/в	7	---
9	sub edx, f	1	н/в	н/в	н/в	---
10	sub eax, edx	-22	н/в	н/в	н/в	---
11	mov res_asm, eax	-22	н/в	н/в	н/в	---

**Висновок:** Ми набули навичок роботи з одновимірними масивами.

		Абанін К.А			ЖДТУ.18.121.01.000 – Лр4	Арк.
		Байлюк Є.М				5
Змн.	Арк.	№ докум.	Підпис	Дата		