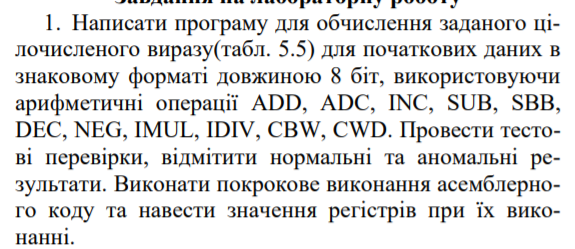
**ЛАБОРАТОРНА РОБОТА № 4**

***Мета:*** 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:**

Завдання 1:





#include <stdio.h>

#include <cstdlib>

#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 = (2 + c - d \* 23 - e) / (2 \* a \* a - 1 - f);

printf("RESULT C=%d\n", res\_c);

\_asm

{

mov al, d;

mov bl, 23;

imul bl;

neg al;

mov cl, c;

add al, 2;

add al, cl;

sub al, e;//<al>=(2+c-d\*23-e)

mov cl, al;

cbw;

mov al, 2;

mov bl, a;

imul bl;

imul bl;

sub al, 1;

sub al, f;

mov bl, al;

mov al, cl;

cbw;

idiv bl;

mov res\_asm, al;

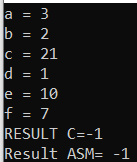
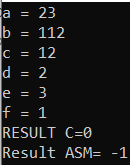
}

printf("Result ASM= %d\n", res\_asm);

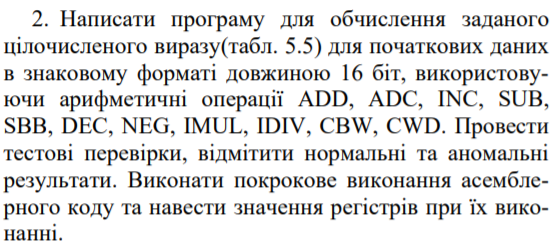
system("Pause"); return 0;

}

**Результат виконання:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Крок | Команда | Значення регістра | | | |
| al | bl | cl | <al:ah> |
| 1 | mov al, d; | 1 | - | - | - |
| 2 | mov bl, 23; | - | 23 | - | - |
| 3 | imul bl; | 23 | 23 | - | 23 |
| 4 | neg al | -23 | - | - | - |
| 5 | mov cl, c; | - | - | 21 | - |
| 6 | add al, 2 | -21 | - | - | - |
| 7 | add al, cl | 0 | - | - | - |
| 8 | sub al, e | -10 | - | - | - |
| 9 | mov cl, al; | - | - | -10 | - |
| 10 | cbw; | - | - | -10 | -10 |
| 11 | mov al, 2; | 2 | - | - | - |
| 12 | mov bl, a; | - | 3 | - | - |
| 13 | Imul bl; | 6 | - | - | 6 |
| 14 | imul bl; | 18 | - | - | 18 |
| 15 | sub al, 1; | 17 | - | - | - |
| 16 | sub al, f; | 10 | - | - | - |
| 17 | mov bl, al; | - | 10 | - | - |
| 18 | mov al, cl; | -10 | - | - | - |
| 19 | cbw; | 10 | - | - | 10 |
| 20 | idiv bl; | -1 | - | - | -1 |





#include <stdio.h>

#include <Windows.h>

int main() {

short int a, f, c, d, e,b, res, res\_asm;

printf("a = ");

scanf\_s("%hd", &a);

printf("b = ");

scanf\_s("%hd", &b);

printf("c = ");

scanf\_s("%hd", &c);

printf("d = ");

scanf\_s("%hd", &d);

printf("e = ");

scanf\_s("%hd", &e);

printf("f = ");

scanf\_s("%hd", &f);

res = (2 + c - d \* 23 - e) / (2 \* a \* a - 1 - f);

\_\_asm

{

mov ax, d;

mov bx, 23;

imul bx;

neg ax;

mov cx, c;

add ax, 2;

add ax, cx;

sub ax, e;//<al>=(2+c-d\*23-e)

mov cx, ax;

cwd;

mov ax, 2;

mov bx, a;

imul bx;

imul bx;

sub ax, 1;

sub ax, f;

mov bx, ax;

mov ax, cx;

cwd;

idiv bx;

mov res\_asm, ax;

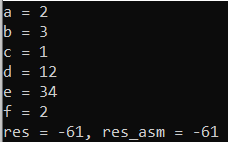
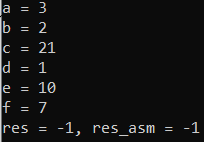
}

printf("res = %d, res\_asm = %d\n", res, res\_asm);

system("pause");

return 0;

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Крок | Команда | Значення регістра | | | |
| ax | bx | cx | <ax:ah> |
| 1 | mov ax, d; | 1 | - | - | - |
| 2 | mov bx, 23; | - | 23 | - | - |
| 3 | imul bx; | 23 | 23 | - | 23 |
| 4 | neg ax | -23 | - | - | - |
| 5 | mov cx, c; | - | - | 21 | - |
| 6 | add ax, 2 | -21 | - | - | - |
| 7 | add ax, cx | 0 | - | - | - |
| 8 | sub ax, e | -10 | - | - | - |
| 9 | mov cx, ax; | - | - | -10 | - |
| 10 | cwd; | - | - | -10 | -10 |
| 11 | mov ax, 2; | 2 | - | - | - |
| 12 | mov bx, a; | - | 3 | - | - |
| 13 | Imul bx; | 6 | - | - | 6 |
| 14 | imul bx; | 18 | - | - | 18 |
| 15 | sub ax, 1; | 17 | - | - | - |
| 16 | sub ax, f; | 10 | - | - | - |
| 17 | mov bx, ax; | - | 10 | - | - |
| 18 | mov ax, cx; | -10 | - | - | - |
| 19 | cwd; | 10 | - | - | 10 |
| 20 | idiv bx; | -1 | - | - | -1 |

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 <stdio.h>

#include <Windows.h>

int main() {

int a, f, c, d, e,b, res, res\_asm;

printf("a = ");

scanf\_s("%hd", &a);

printf("b = ");

scanf\_s("%hd", &b);

printf("c = ");

scanf\_s("%hd", &c);

printf("d = ");

scanf\_s("%hd", &d);

printf("e = ");

scanf\_s("%hd", &e);

printf("f = ");

scanf\_s("%hd", &f);

res = (2 + c - d \* 23 - e) / (2 \* a \* a - 1 - f);

\_\_asm

{

mov eax, d;

mov ebx, 23;

imul ebx;

neg eax;

mov ecx, c;

add eax, 2;

add eax, ecx;

sub eax, e;//<al>=(2+c-d\*23-e)

mov ecx, eax;

cdq;

mov eax, 2;

mov ebx, a;

imul ebx;

imul ebx;

sub eax, 1;

sub eax, f;

mov ebx, eax;

mov eax, ecx;

cdq;

idiv ebx;

mov res\_asm, eax;

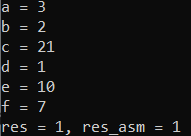
}

printf("res = %d, res\_asm = %d\n", res, res\_asm);

system("pause");

return 0;

}

******

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Крок | Команда | Значення регістра | | | |
| eax | ebx | ecx | <eax:ah> |
| 1 | mov eax, d; | 1 | - | - | - |
| 2 | mov ebx, 23; | - | 23 | - | - |
| 3 | imul ebx; | 23 | 23 | - | 23 |
| 4 | neg eax | -23 | - | - | - |
| 5 | mov ecx, c; | - | - | 21 | - |
| 6 | add eax, 2 | -21 | - | - | - |
| 7 | add eax, ecx | 0 | - | - | - |
| 8 | sub eax, e | -10 | - | - | - |
| 9 | mov ecx, eax; | - | - | -10 | - |
| 10 | cdq; | - | - | -10 | -10 |
| 11 | mov eax, 2; | 2 | - | - | - |
| 12 | mov ebx, a; | - | 3 | - | - |
| 13 | imul ebx; | 6 | - | - | 6 |
| 14 | imul ebx; | 18 | - | - | 18 |
| 15 | sub eax, 1; | 17 | - | - | - |
| 16 | sub eax, f; | 10 | - | - | - |
| 17 | mov ebx, eax; | - | 10 | - | - |
| 18 | mov eax, ecx; | -10 | - | - | - |
| 19 | cdq; | 10 | - | - | 10 |
| 20 | idiv ebx; | -1 | - | - | -1 |

***Висновки:*** Ми ознайомилися з основними командами мови Assembler для обчислення складних цілочисельних виразів; набули практичних навичок в написанні програм для обчислення складних цілочисельних виразів на мові Assembler.