**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**

з дисципліни  
«Технології розроблення системних програм»

на тему  
«Арифметичні і логічні операції з цілими числами. Масиви.»

Виконав: Перевірив:

студент групи ІП-84 доц. кафедри ОТ  
Голубов Іван Олегович Павлов В. Г.  
номер залікової книжки: 8404

Київ 2020

**Мета роботи**

Вивчення арифметичних і логічних команд Асемблера і здобуття навиків виконання розрахунків з елементами масивів.

**Порядок виконання роботи**

1. Вивчити арифметичні і логічні команди цілочисельної арифметики.

2. Розробити програму на мові Асемблер, в якій згідно з індивідуальним варіантом завдання виконуються обчислення значення арифметичного виразу (c/d + 3\*a/2)/(c - a + 1) з подальшим виводом результату у віконному інтерфейсі.

3. В разі парного результату він перед виводом додатково ділиться на 2, а в разі непарного – результат додатково умножається на 5.

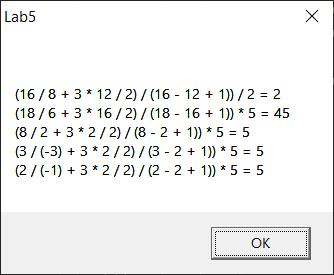
4. Розрахунки (п. 2, 3) повторити в програмі для 5 значень змінних\*\*, причому всі вихідні значення задати цілими числами у вигляді одновимірних масивів.

5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі.

6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.

7. Зробити висновки по лабораторній роботі.

**Скріншоти**



**Лістинг програми**

 .386

.model flat,stdcall

option casemap:none

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

include C:\masm32\include\user32.inc

includelib C:\masm32\lib\kernel32.lib

includelib C:\masm32\lib\user32.lib

.data

    arr\_a        dd    2, 2, 2, 16, 12

    arr\_c        dd    2, 3, 8, 18, 16;

    arr\_d        dd    -1, -3, 2, 6, 8;

    var\_a        dd    ?;

    var\_c        dd    ?;

    var\_d        dd    ?;

    bu1          dd    ?;

    iterator     dd    5;

    const1       dd    3;

    const2       dd    2;

    const3       dd    5;

    head         db    "Lab5", 0;

    format1      db    "(%u / ", 0;

    format2p     db    "%u + 3 \* ", 0;

    format2m     db    "(-%u) + 3 \* ", 0;

    format3      db    "%u / 2) / (", 0;

    format4      db    "%u - ", 0;

    format5      db    "%d + 1)) / 2", 0;

    format7      db    "%d + 1)) \* 5", 0;

    format6      db    " = %u", 0;

    buff1        db    15 DUP (?);

    buff2        db    15 DUP (?);

    buff3        db    15 DUP (?);

    buff4        db    15 DUP (?);

    buff5        db    15 DUP (?);

    buff6        db    15 DUP (?);

    lineFormat   db    6 DUP("%s"), 0;

    format       db    "%s",13;

    lineBuff     db    50 DUP(?), 0;

    buff         db    ?;

.code

start:

   xor ebx, ebx;

    startLoop:

        mov ebx, iterator;

        dec ebx;

        mov eax, arr\_a[ebx \* 4];

        mov var\_a, eax;

        mov eax, arr\_c[ebx \* 4];

        mov var\_c, eax;

        mov eax, arr\_d[ebx \* 4];

        mov var\_d, eax;

        xor eax, eax

        mov eax, var\_a;

        mul const1;    3\*a

        div const2;    3\*a/2

        mov bu1, eax ;bu1=3\*a/2

        add var\_d, 0;

        ; sign check

        js negative

            invoke wsprintf, addr buff2, addr format2p, var\_d;

            mov edx, 0;

            mov eax, var\_c;

            mov ecx, var\_d;

            div ecx;       c/d

            add eax, bu1; c/d+3\*a/2

            jmp positive

        negative:

            neg var\_d

            invoke wsprintf, addr buff2, addr format2m, var\_d;

            mov edx, 0;

            mov eax, var\_c;

            mov ecx, var\_d;

            div ecx;       c/d

            sub bu1, eax;   c/d+3\*a/2

            ; sub because var\_d is neg

            mov eax, bu1 ; eax=c/d+3\*a/2

        positive:

        xor ecx, ecx

        xor edx, edx

        mov ecx, var\_c

        inc ecx; ecx = c+1

        mov edx, var\_a

        sub ecx, edx ; ecx = c-d+1

        mov edx, 0

        mov eax, eax

        mov ecx, ecx

        div ecx;eax=(c/d+3\*a/2)/(c-d+1)

        test eax, 1;parity check

        jz evenNumb

        jnz unevenNumb

        unevenNumb:

            xor ecx, ecx

            mov eax, eax;

            mov ecx, const3; const3=5

            mul ecx; ecx=ecx\*5

            invoke wsprintf, addr buff6, addr format6, eax;

            invoke wsprintf, addr buff5, addr format7, var\_a;

            jmp Numb

        evenNumb:

            xor ecx, ecx

            mov edx, 0

            mov eax, eax

            mov ecx, const2; const2=2

            div ecx; ecx=ecx/2

            invoke wsprintf, addr buff6, addr format6, eax;

            invoke wsprintf, addr buff5, addr format5, var\_a;

        Numb:

        invoke wsprintf, addr buff1, addr format1, var\_c;

        invoke wsprintf, addr buff3, addr format3, var\_a;

        invoke wsprintf, addr buff4, addr format4, var\_c;

        invoke wsprintf, addr lineBuff,

                         addr lineFormat,

                         addr buff1,

                         addr buff2,

                         addr buff3,

                         addr buff4,

                         addr buff5,

                         addr buff6;

        invoke wsprintf, addr buff, addr format, addr buff, addr lineBuff;

    dec iterator;

    jnz startLoop

    invoke MessageBox,NULL, addr buff, addr head, MB\_OK;

    invoke ExitProcess, NULL;

end start

**Перевірка результатів розрахунків**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | a=2  c=2  d=-1 | a=2  c=3  d=-3 | a=2  c=8  d=2 | a=16  c=18  d=6 | a=12  c=16  d=8 |
| 3\*a | 6 | 6 | 6 | 48 | 36 |
|  | 3 | 3 | 3 | 24 | 18 |
|  | -2 | -1 | 4 | 3 | 2 |
|  | 1 | 2 | 7 | 27 | 20 |
| c+1 | 3 | 4 | 9 | 19 | 17 |
| c-a+1 | 1 | 2 | 7 | 3 | 5 |
|  | 1 | 1 | 1 | 9 | 4 |
| Перевірка на парність |  | | | | |
| Парне/2 |  |  |  |  | 2 |
| Непарне\*5 | 5 | 5 | 5 | 45 |  |
| Контрольний  результат | 5 | 5 | 5 | 45 | 2 |
| Резутат програми | 5 | 5 | 5 | 45 | 2 |
| Коректність роботи програми | ˅ | ˅ | ˅ | ˅ | ˅ |

**Висновки**

У ході виконання лабораторної роботи я вивчив арифметичні і логічні команд Асемблера і здобув навики виконання розрахунків з елементами масивів

Я створив програму на мові асемблер, в якій згідно з індивідуальним варіантом завдання виконуються обчислення значення арифметичного виразу (c/d + 3\*a/2)/(c - a + 1) з подальшим виводом результату у віконному інтерфейсі. Програма в разі парного результату перед виводом додатково ділить його на 2, а в разі непарного – результат додатково умножається на 5.

Повторив для 5 значень змінних, причому всі вихідні значення задати цілими числами у вигляді одновимірних масивів.

Перевірив правильність виконання розрахунків і результатів, що виводяться, заздалегідь виконав контрольні розрахунки. Виконав відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Проміжні і остаточні результати контрольних розрахунків привів в таблиці.