**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 №6**

з дисципліни  
«Технології розроблення системних програм»

на тему  
«Обчислення арифметичних виразів і трансцендентних функцій.

Використання команд співпроцесора ix87.»

Виконав: Перевірив:

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

Київ 2020

**Мета роботи**

Вивчення команд Асемблера для арифметики з плаваючою комою і здобуття навичок виконання розрахунків з елементами масивів.

**Порядок виконання роботи**

1. Вивчити арифметичні команди з плаваючою комою.

2. Розробити програму на мові Асемблер, в якій згідно з індивідуальним варіантом завдання виконуються обчислення значення арифметичного вираження із застосуванням команд співпроцесора ix87 з подальшим виведенням результату\* у віконному інтерфейсі.

3. Для всіх варіантів передбачити завдання значень вхідних змінних у форматі double (DQ), а результату обчислень – у форматі long double (DT).

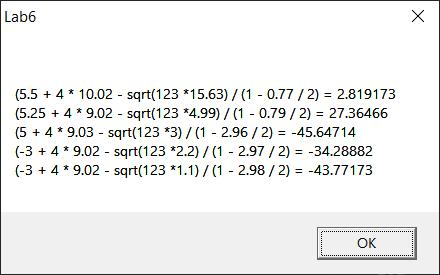
4. Розрахунки (п. 2, 3) повторити в програмі для 5 значень змінних\*\*, причому всі вхідні значення задати дійсними числами у вигляді одновимірних масивів.

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**

.686

.model flat,stdcall

option casemap:none

include C:\masm32\include\windows.inc      ;MB\_OK,NULL

include C:\masm32\include\kernel32.inc     ;ExitProcess

include C:\masm32\include\user32.inc       ;wsprintf, MessageBox

include C:\masm32\include\masm32.inc       ;FloatToStr

include C:\masm32\include\fpu.inc          ;FpuFLtoA

includelib C:\masm32\lib\kernel32.lib      ;ExitProcess

includelib C:\masm32\lib\user32.lib        ;MessageBox

includelib C:\masm32\lib\masm32.lib        ;FloatToStr

includelib C:\masm32\lib\fpu.lib           ;FpuFLtoA

.data

    arr\_a        dq     2.98,  2.97,  2.96,  0.79,   0.77;

    arr\_b        dq     9.02,  9.02,  9.03,  9.02,  10.02;

    arr\_c        dq     -3.0,  -3.0,   5.0,  5.25,    5.5;

    arr\_d        dq      1.1,   2.2,   3.0,  4.99,  15.63;

    var\_a        dq    ?;

    var\_b        dq    ?;

    var\_c        dq    ?;

    var\_d        dq    ?;

    printRes     dt    0;

    iterator     dw    5;

    buff1        db    25 DUP (?);

    buff2        db    25 DUP (?);

    buff3        db    25 DUP (?);

    buff4        db    25 DUP (?);

    buff5        db    25 DUP (?);

    buff         DW    250 DUP(?), 0;

    lineFormat   db    5 DUP("%s"), 0;

    format       db    "%s",13;

    lineBuff     DW    50 DUP(?), 0;

.const

    const1       dq    4.0;

    const2       dq    123.0;

    const3       dq    2.0;

    head         db    "Lab6", 0;

    format1      db    "(%s + 4 \* %s - sqrt(123 \* %s)) / (1 - %s / 2) = %s ", 0;

.code

start:

    finit

    startLoop:

        xor eax, eax;

        xor ebx, ebx;

        mov bx, iterator;

        dec bx;

        fld arr\_a[ebx \* 8];

        fstp var\_a;

        fld arr\_b[ebx \* 8];

        fstp var\_b;

        fld arr\_c[ebx \* 8];

        fstp var\_c;

        fld arr\_d[ebx \* 8];

        fstp var\_d;

        fld var\_b;              ST(0) <- b

        fmul const1;            ST(0) <- 4\*b

        fadd var\_c;             ST(0) <- c+4\*b

        fld var\_d;              ST(1) <- c+4\*b ST(0) <- d

        fmul const2;            ST(1) <- c+4\*b ST(0) <- 123\*d

        fsqrt;                  ST(1) <- c+4\*b ST(0) <- sqrt(123\*d)

        fsub;                   ST(0) <- c+4\*b - sqrt(123\*d)

        fld1;                   ST(1) <- c+4\*b - sqrt(123\*d)

        fld var\_a;              ST(2) <- c+4\*b - sqrt(123\*d), ST(1) <- 1, ST(0) <- a

        fld const3;             ST(3) <- c+4\*b - sqrt(123\*d), ST(2) <- 1, ST(1) <- a, ST(0) <-2

        fdiv;                   ST(2) <- c+4\*b - sqrt(123\*d), ST(1) <- 1, ST(0) <- a/2

        fsub;                   ST(1) <- c+4\*b - sqrt(123\*d), ST(0) <- 1-a/2

        fdiv;                   ST(0) <- (c+4\*b- sqrt(123\*d))/(1- a/2)

        fstp printRes;          printRes <- ST(0)

        invoke FloatToStr, var\_c, addr buff1;

        invoke FloatToStr, var\_b, addr buff2;

        invoke FloatToStr, var\_d, addr buff3;

        invoke FloatToStr, var\_a, addr buff4;

        invoke FpuFLtoA,offset  printRes,8,offset  buff5, SRC1\_REAL

        invoke wsprintf, addr lineBuff,

                         addr format1,

                         addr buff1,

                         addr buff2,

                         addr buff3,

                         addr buff4,

                         addr buff5;

        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=2.98  b=9.02  c=-3  d=1.1 | a=2.97  b=9.02  c=-3  d=2.2 | a=2.96  b=9.03  c=5  d=3 | a=0.79  b=9.02  c=5.25  d=4.99 | a=0.77  b=10.02  c=5.5  d=15.63 |
| 4\**b* | 36.08 | 36.08 | 36.12 | 36.08 | 40.08 |
| *c*+4\**b* | 33.08 | 33.08 | 41.12 | 41.33 | 45.58 |
|  | 135.3 | 270.6 | 369 | 613.77 | 1922.49 |
|  | 11.632 | 16.45 | 19.209 | 24.774 | 43.846 |
|  | 21.448 | 16.63 | 21.911 | 16.556 | 1.734 |
|  | 1.49 | 1.485 | 1.48 | 0.395 | 0.385 |
|  | -0.49 | -0.485 | -0.48 | 0.605 | 0.615 |
|  | -43.771 | -34.289 | -45.648 | 27.365 | 2.815 |
| Резутат програми | -43.77173 | -34.28882 | -45.64714 | 27.36466 | 2.819173 |
| Коректність роботи програми | ˅ | ˅ | ˅ | ˅ | ˅ |

**Висновки**

У ході виконання лабораторної роботи я вивчив команди Асемблера для арифметики з плаваючою комою і здобув навички виконання розрахунків з елементами масивів

Я створив програму на мові асемблер, в якій згідно з індивідуальним варіантом завдання виконуються обчислення значення арифметичного виразу  із застосуванням команд співпроцесора ix87 з подальшим виводом результату у віконному інтерфейсі. Програма має вхідні змінні у форматі double (DQ), а результат обчислень – у форматі long double (DT).Повторив для 5 значень змінних, причому всі вихідні значення задав дійсними 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.