**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), проміжних результатів обчислень – у форматі long double (DT), а кінцевих - знову у форматі double.

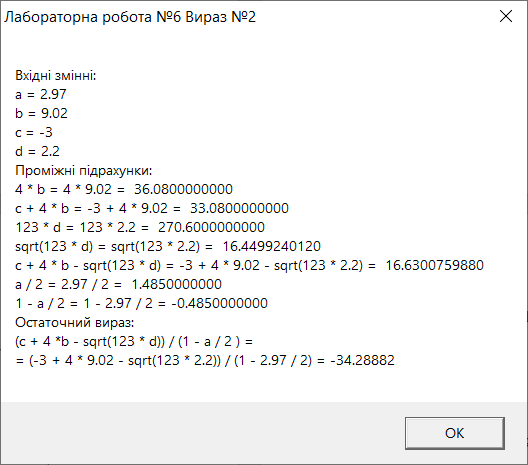
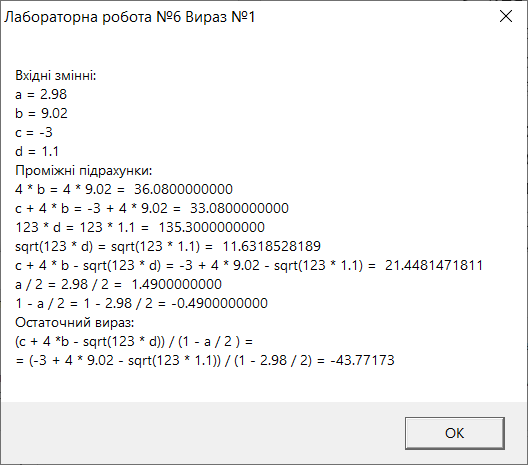
4. Розрахунки (п. 2, 3) повторити в програмі для 5 значень змінних\*\*, причому всі вхідні значення задати дійсними числами у вигляді одновимірних масивів.

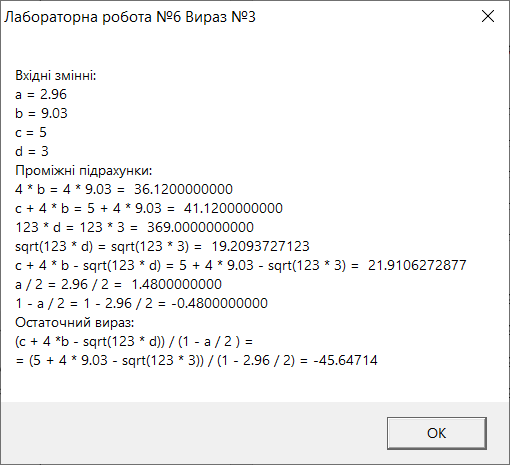
5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі.

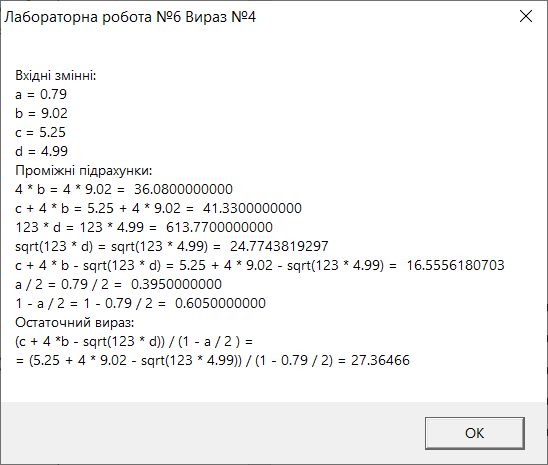
6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.

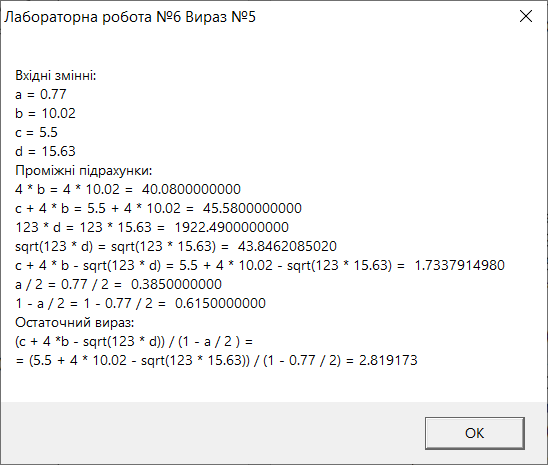
7. Зробити висновки по лабораторній роботі.

**Скріншоти**









**Лістинг програми**

.686

.model flat,stdcall

option casemap:none

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

include \masm32\include\kernel32.inc ;ExitProcess

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

include \masm32\include\masm32.inc ;FloatToStr

include \masm32\include\fpu.inc ;FpuFLtoA

includelib \masm32\lib\kernel32.lib ;ExitProcess

includelib \masm32\lib\user32.lib ;MessageBox

includelib \masm32\lib\masm32.lib ;FloatToStr

includelib \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 dq ?; В

intermedRes1 DT ?;

intermedRes2 DT ?;

intermedRes3 DT ?;

intermedRes4 DT ?;

intermedRes5 DT ?;

intermedRes6 DT ?;

intermedRes7 DT ?;

RES DT ?;

iter dw 1;

iterator dw -1;

buffa db 25 DUP (?);

buffb db 25 DUP (?);

buffc db 25 DUP (?);

buffd db 25 DUP (?);

buff8 db 250 DUP (?);

buff1 db 250 DUP (?);

buff2 db 250 DUP (?);

buff3 db 250 DUP (?);

buff4 db 250 DUP (?);

buff5 db 250 DUP (?);

buff6 db 250 DUP (?);

buff7 db 250 DUP (?);

buff9 db 250 DUP (?);

buff db 1000 DUP(?), 0;

mainbuff db 1000 DUP(?), 0;

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

format db "%s",13,"%s",13, "%s",13, "%s",13, "%s",13, "%s",13, "%s",13, "%s",13

mainformat db "%s",13

li DW 50 DUP(?), 0;

lineBuff1 DW 50 DUP(?), 0;

lineBuff2 DW 50 DUP(?), 0;

lineBuff3 DW 50 DUP(?), 0;

lineBuff4 DW 50 DUP(?), 0;

lineBuff5 DW 50 DUP(?), 0;

lineBuff6 DW 50 DUP(?), 0;

lineBuff7 DW 50 DUP(?), 0;

lineBuff8 DW 50 DUP(?), 0;

lineBuff9 DW 100 DUP(?), 0;

count db 50 DUP(?);

.const

const1 dq 4.0;

const2 dq 123.0;

const3 dq 2.0;

head db "Лабораторна робота №6 Вираз №%d", 0;

headtask db "Task N", 0;

head1 db "Вхідні змінні:" , 0;

head2 db "Проміжні підрахунки:", 0;

head3 db "Остаточний вираз:", 0;

head4 db "(c + 4 \*b - sqrt(123 \* d)) / (1 - a / 2 ) =", 0;

letter db "%s",13, "a = %s",13,"b = %s", 13, "c = %s", 13,"d = %s", 0;

forma db "%s", 0;

format1 db "%s",13, "4 \* b = 4 \* %s = %s", 0

format2 db "c + 4 \* b = %s + 4 \* %s = %s", 0;

format3 db "123 \* d = 123 \* %s = %s", 0;

format4 db "sqrt(123 \* d) = sqrt(123 \* %s) = %s", 0;

format5 db "c + 4 \* b - sqrt(123 \* d) = %s + 4 \* %s - sqrt(123 \* %s) = %s", 0;

format6 db "a / 2 = %s / 2 = %s", 0;

format7 db "1 - a / 2 = 1 - %s / 2 = %s", 0;

format8 db "%s",13, "%s",13,"= (%s + 4 \* %s - sqrt(123 \* %s)) / (1 - %s / 2) = %s ", 0;

.code

start:

finit

startLoop:

xor ebx, ebx;

mov bx, iterator;

inc 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

fstp intermedRes1

fld intermedRes1

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

fstp intermedRes2

fld intermedRes2

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

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

fstp intermedRes3

fld intermedRes3

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

fstp intermedRes4;

fld intermedRes4

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

fstp intermedRes5

fld intermedRes5

fld1; ST(1) <- c+4\*b - sqrt(123\*d), ST(0) <- 1

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

fstp intermedRes6

fld intermedRes6

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

fstp intermedRes7

fld intermedRes7

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

fstp printRes; DQ printRes <- ST(0)

fld printRes

fstp RES;

invoke wsprintf, addr count, addr head, iter;

invoke FloatToStr, var\_c, addr buffc;

invoke FloatToStr, var\_b, addr buffb;

invoke FloatToStr, var\_d, addr buffd;

invoke FloatToStr, var\_a, addr buffa;

invoke FloatToStr, printRes, addr buff8;

invoke FpuFLtoA,offset RES, 50,offset buff8, SRC1\_REAL

invoke FpuFLtoA,offset intermedRes1, 50,offset buff1, SRC1\_REAL

invoke FpuFLtoA,offset intermedRes2, 50,offset buff2, SRC1\_REAL

invoke FpuFLtoA,offset intermedRes3, 50,offset buff3, SRC1\_REAL

invoke FpuFLtoA,offset intermedRes4, 50,offset buff4, SRC1\_REAL

invoke FpuFLtoA,offset intermedRes5, 50,offset buff5, SRC1\_REAL

invoke FpuFLtoA,offset intermedRes6, 50,offset buff6, SRC1\_REAL

invoke FpuFLtoA,offset intermedRes7, 50,offset buff7, SRC1\_REAL

invoke wsprintf, addr lineBuff1,

addr letter,

addr head1,

addr buffa,

addr buffb,

addr buffc,

addr buffd

invoke wsprintf, addr lineBuff2,

addr format1,

addr head2,

addr buffb,

addr buff1

invoke wsprintf, addr lineBuff3,

addr format2,

addr buffc,

addr buffb,

addr buff2;

invoke wsprintf, addr lineBuff4,

addr format3,

addr buffd,

addr buff3;

invoke wsprintf, addr lineBuff5,

addr format4,

addr buffd,

addr buff4;

invoke wsprintf, addr lineBuff6,

addr format5,

addr buffc,

addr buffb,

addr buffd,

addr buff5;

invoke wsprintf, addr lineBuff7,

addr format6,

addr buffa,

addr buff6;

invoke wsprintf, addr lineBuff8,

addr format7,

addr buffa,

addr buff7;

invoke wsprintf, addr lineBuff9,

addr format8,

addr head3,

addr head4,

addr buffc,

addr buffb,

addr buffd,

addr buffa,

addr buff8

invoke wsprintf, addr buff,

addr format,

addr lineBuff1,

addr lineBuff2,

addr lineBuff3,

addr lineBuff4,

addr lineBuff5,

addr lineBuff6,

addr lineBuff7,

addr lineBuff8,

addr lineBuff9;

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

inc iter;

inc iterator;

cmp iterator, 4

jnz startLoop

invoke ExitProcess, NULL;

end start

**Перевірка результатів розрахунків**

Похибки обчислень виділені **жирним шрифтом**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | №1 | №2 | №3 |
| a=0.77  b=10.02  c=5.5  d=15.63 | a=0.79  b=9.02  c=5.25  d=4.99 | a=2.96  b=9.03  c=5  d=3 |
| 4\**b* | Власні розрахунки | 40.08 | 36.08 | 36.12 |
| Програма | 40.08 | 36.08 | 36.12 |
| *c*+4\**b* | Власні розрахунки | 45.58 | 41.33 | 41.12 |
| Програма | 45.58 | 41.33 | 41.12 |
|  | Власні розрахунки | 1922.49 | 613.77 | 369 |
| Програма | 1922.49 | 613.77 | 369 |
|  | Власні розрахунки | 43.8462085**02** | 24.7743819**2** | 19.2093727**1** |
| Програма | 43.8462085 | 24.7743819 | 19.2093727 |
|  | Власні розрахунки | 1.733791**49** | 16.555618**07** | 21.910627**28** |
| Програма | 1.733791**5** | 16.555618**1** | 21.910627**3** |
|  | Власні розрахунки | 0.385 | 0.395 | 1.48 |
| Програма | 0.385 | 0.395 | 1.48 |
|  | Власні розрахунки | 0.615 | 0.605 | -0.48 |
| Програма | 0.615 | 0.605 | -0.48 |
|  | Власні розрахунки | 2.819173**1** | 27.3646**57** | -45.64714**01** |
| Програма | 2.819173 | 27.3646**6** | -45.64714 |
| Коректність роботи програми |  | ˅ | ˅ | ˅ |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | №4 | №5 |
| a=2.97  b=9.02  c=-3  d=2.2 | a=2.98  b=9.02  c=-3  d=1.1 |
| 4\**b* | Власні розрахунки | 36.08 | 36.08 |
| Програма | 36.08 | 36.08 |
| *c*+4\**b* | Власні розрахунки | 33.08 | 33.08 |
| Програма | 33.08 | 33.08 |
|  | Власні розрахунки | 270.6 | 135.3 |
| Програма | 270.6 | 135.3 |
|  | Власні розрахунки | 16.4499240**1** | 11.6318528**1** |
| Програма | 16.4499240 | 11.6318528 |
|  | Власні розрахунки | 16.63007**598** | 21.448147**18** |
| Програма | 16.63007**60** | 21.448147**2** |
|  | Власні розрахунки | 1.485 | 1.49 |
| Програма | 1.485 | 1.49 |
|  | Власні розрахунки | -0.485 | -0.49 |
| Програма | -0.485 | -0.49 |
|  | Власні розрахунки | -34.2888**16** | -43.7717**28** |
| Програма | -34.2888**2** | -43.7717**3** |
| 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.

У зв’язку з використанням FloatToStr і FpuFLtoA точність при виводі результату на екран була зменшена даними командами.

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. Програма з заданою точністю змогла чітко розрахувати вираз і зобразити його на екрані з підстановкою значень і результату.