**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.

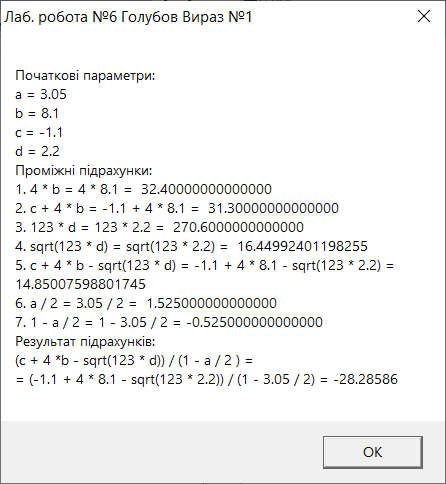
4. Розрахунки (п. 2, 3) повторити в програмі для 5 значень змінних\*\*, причому всі вхідні значення задати дійсними числами у вигляді одновимірних масивів.

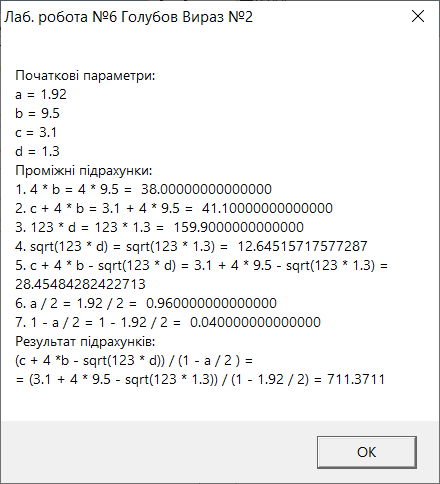
5. Для перевірки правильності виконання розрахунків і результатів, що виводяться, заздалегідь виконати контрольні розрахунки. Проміжні і остаточні результати контрольних розрахунків привести в звіті по лабораторній роботі.

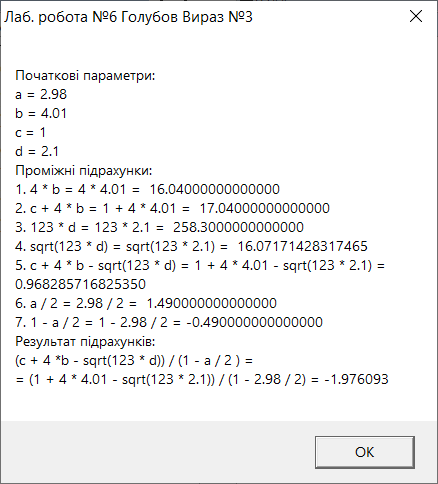
6. Виконати відладку програми шляхом порівняння розрахованих програмою результатів з контрольними прикладами. Лістинг розробленої програми і скріншоти розрахунків по всіх контрольних прикладах привести в звіті по лабораторній роботі.

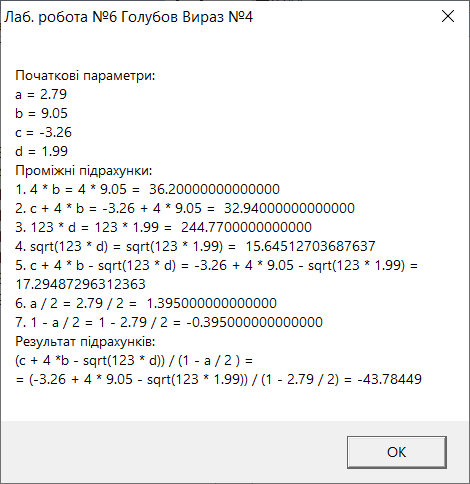
7. Зробити висновки по лабораторній роботі.

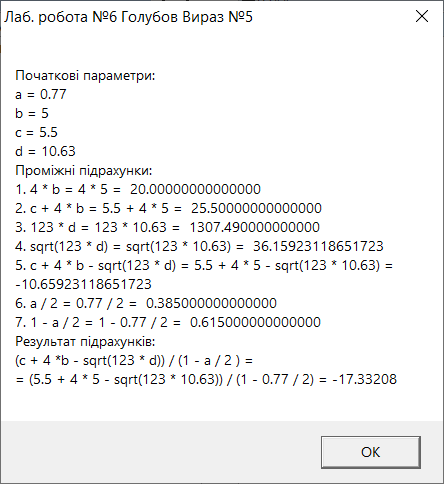
**Скріншоти**



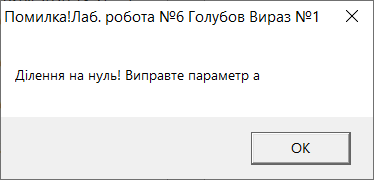




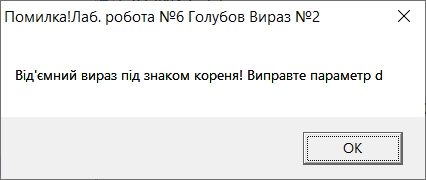




**Скріншот роботи програми при а=2**



**Скріншот роботи програми при d<0**



**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

var\_a DQ ?;

var\_b DQ ?;

var\_c DQ ?;

var\_d DQ ?;

Res DQ ?;

temp1 DT ?;

temp2 DT ?;

temp3 DT ?;

temp4 DT ?;

temp5 DT ?;

temp6 DT ?;

temp7 DT ?;

RES DT ?;

it DW 1;

iter DW 0;

arr\_a DQ 3.05, 1.92, 2.98, 2.79, 0.77;

arr\_b DQ 8.10, 9.50, 4.01, 9.05, 5.0;

arr\_c DQ -1.1, 3.1, 1.0, -3.26, 5.5;

arr\_d DQ 2.2, 1.3, 2.1, 1.99, 10.63;

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;

linetemplate DB 5 DUP("%s"), 0;

template DB "%s",13,"%s",13, "%s",13, "%s",13, "%s",13, "%s",13, "%s",13, "%s",13

maintemplate DB "%s",13

li DW 50 DUP(?), 0;

line1 DW 50 DUP(?), 0;

line2 DW 50 DUP(?), 0;

line3 DW 50 DUP(?), 0;

line4 DW 50 DUP(?), 0;

line5 DW 50 DUP(?), 0;

line6 DW 50 DUP(?), 0;

line7 DW 50 DUP(?), 0;

line8 DW 50 DUP(?), 0;

line9 DW 100 DUP(?), 0;

counter DB 50 DUP(?);

.const

var1 DQ 4.0;

var2 DQ 123.0;

var3 DQ 2.0;

var4 DQ 0.0;

head DB "Лаб. робота №6 Голубов Вираз №%d", 0;

headerr DB "Помилка!Лаб. робота №6 Голубов Вираз №%d",0

text1 DB "Від'ємний вираз під знаком кореня! Виправте параметр d", 0;

text2 DB "Ділення на нуль! Виправте параметр a", 0;

headtask DB "N", 0;

head1 DB "Початкові параметри:" , 0;

head2 DB "Проміжні підрахунки:", 0;

head3 DB "Результат підрахунків:", 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;

formate DB "%s", 0;

template1 DB "%s",13, "1. 4 \* b = 4 \* %s = %s", 0

template2 DB "2. c + 4 \* b = %s + 4 \* %s = %s", 0;

template3 DB "3. 123 \* d = 123 \* %s = %s", 0;

template4 DB "4. sqrt(123 \* d) = sqrt(123 \* %s) = %s", 0;

template5 DB "5. c + 4 \* b - sqrt(123 \* d) = %s + 4 \* %s - sqrt(123 \* %s) = %s", 0;

template6 DB "6. a / 2 = %s / 2 = %s", 0;

template7 DB "7. 1 - a / 2 = 1 - %s / 2 = %s", 0;

template8 DB "%s",13, "%s",13,"= (%s + 4 \* %s - sqrt(123 \* %s)) / (1 - %s / 2) = %s ", 0;

.code

mainStart:

finit

mainLoop:

xor ebx, ebx;

mov bx, iter;

fld arr\_a[ebx \* 8];

fcom var3

fstsw ax

sahf

jz eror1

xor eax, eax

jmp first

eror1:

invoke wsprintf, addr counter, addr headerr, iter;

invoke MessageBox,NULL, addr text1, addr counter, MB\_OK;

jmp ending

first:

fstp var\_a;

fld arr\_b[ebx \* 8];

fstp var\_b;

fld arr\_c[ebx \* 8];

fstp var\_c;

fld arr\_d[ebx \* 8];

fcom var4

fstsw ax

sahf

jc eror2

xor eax, eax

jmp second

eror2:

invoke wsprintf, addr counter, addr headerr, iter;

invoke MessageBox,NULL, addr text2, addr counter, MB\_OK;

jmp ending

second:

fstp var\_d;

fld var\_b;

fmul var1;

fstp temp1

fld temp1

fadd var\_c;

fstp temp2

fld temp2

fld var\_d;

fmul var2;

fstp temp3

fld temp3

fsqrt;

fstp temp4;

fld temp4

fsub;

fstp temp5

fld temp5

fld1;

fld var\_a;

fld var3;

fdiv;

fstp temp6

fld temp6

fsub;

fstp temp7

fld temp7

fdiv;

fstp Res; DQ

invoke wsprintf, addr counter, addr head, it;

invoke FloatToStr, var\_a, addr buffa;

invoke FloatToStr, var\_b, addr buffb;

invoke FloatToStr, var\_c, addr buffc;

invoke FloatToStr, var\_d, addr buffd;

invoke FloatToStr, Res, addr buff8;

invoke FpuFLtoA,offset temp1, 15,offset buff1, SRC1\_REAL

invoke FpuFLtoA,offset temp2, 15,offset buff2, SRC1\_REAL

invoke FpuFLtoA,offset temp3, 15,offset buff3, SRC1\_REAL

invoke FpuFLtoA,offset temp4, 15,offset buff4, SRC1\_REAL

invoke FpuFLtoA,offset temp5, 15,offset buff5, SRC1\_REAL

invoke FpuFLtoA,offset temp6, 15,offset buff6, SRC1\_REAL

invoke FpuFLtoA,offset temp7, 15,offset buff7, SRC1\_REAL

invoke wsprintf, addr line1, addr letter, addr head1, addr buffa, addr buffb, addr buffc, addr buffd

invoke wsprintf, addr line2, addr template1, addr head2, addr buffb, addr buff1

invoke wsprintf, addr line3, addr template2, addr buffc, addr buffb, addr buff2;

invoke wsprintf, addr line4, addr template3, addr buffd, addr buff3;

invoke wsprintf, addr line5, addr template4, addr buffd, addr buff4;

invoke wsprintf, addr line6, addr template5, addr buffc, addr buffb, addr buffd, addr buff5;

invoke wsprintf, addr line7, addr template6, addr buffa, addr buff6;

invoke wsprintf, addr line8, addr template7, addr buffa, addr buff7;

invoke wsprintf, addr line9, addr template8, addr head3, addr head4, addr buffc, addr buffb, addr buffd, addr buffa, addr buff8

invoke wsprintf, addr buff, addr template, addr line1, addr line2, addr line3, addr line4, addr line5, addr line6, addr line7, addr line8, addr line9;

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

jmp ending

ending:

inc it;

inc iter;

cmp iter, 5

jnz mainLoop

invoke ExitProcess, NULL;

end mainStart

**Перевірка результатів розрахунків**

Похибки обчислень виділені **жирним шрифтом**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | №1 | №2 | №3 |
| a=3.05  b=8.1  c=-1.1  d=2.2 | a=1.92  b=9.5  c=3.1  d=1.3 | a=2.98  b=4.01  c=1  d=2.1 |
| 4\**b* | Власні розрахунки | 32.4 | 38 | 16.04 |
| Програма | 32.4 | 38 | 16.04 |
| *c*+4\**b* | Власні розрахунки | 31.3 | 41.1 | 17.04 |
| Програма | 31.3 | 41.1 | 17.04 |
|  | Власні розрахунки | 270.6 | 159.9 | 258.3 |
| Програма | 270.6 | 159.9 | 258.3 |
|  | Власні розрахунки | 16.44992401198255**01** | 12.64515717577287**2** | 16.07171428317465**1** |
| Програма | 16.44992401198255 | 12.64515717577287 | 16.07171428317465 |
|  | Власні розрахунки | 14.85007598801745**2** | 28.45484282422713**01** | 0.9682857168253**498** |
| Програма | 14.85007598801745 | 28.45484282422713 | 0.9682857168253**5** |
|  | Власні розрахунки | 1.525 | 0.96 | 1.49 |
| Програма | 1.525 | 0.96 | 1.49 |
|  | Власні розрахунки | -0.525 | 0.04 | -0.49 |
| Програма | -0.525 | 0.04 | -0.49 |
|  | Власні розрахунки | -28.2858**59** | 711.3711**02** | -1.976093**02** |
| Програма | -28.2858**6** | 711.3711 | -1.976093 |
| Коректність роботи програми |  | ˅ | ˅ | ˅ |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | №4 | №5 |
| a=2.97  b=9.05  c=-3.26  d=1.99 | a=0.77  b=5  c=5.5  d=10.63 |
| 4\**b* | Власні розрахунки | 36.2 | 20 |
| Програма | 36.2 | 20 |
| *c*+4\**b* | Власні розрахунки | 32.94 | 25.5 |
| Програма | 32.94 | 25.5 |
|  | Власні розрахунки | 244.77 | 1307.49 |
| Програма | 244.77 | 1307.49 |
|  | Власні розрахунки | 15.6451270368763**69** | 36.15923118651723**3** |
| Програма | 15.64512703687637 | 36.15923118651723 |
|  | Власні розрахунки | 17.29487296312363**1** | -10.659231186517230**1** |
| Програма | 17.29487296312363 | -10.659231186517230 |
|  | Власні розрахунки | 1.395 | 0.385 |
| Програма | 1.395 | 0.385 |
|  | Власні розрахунки | -0.395 | 0.615 |
| Програма | -0.395 | 0.615 |
|  | Власні розрахунки | -43.78449**1** | -17.3320**78** |
| Програма | -43.78449 | -17.33208 |
| 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.