**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут ім. Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Лабораторна робота № 6**

з дисципліни

«Системне програмування»

на тему

«Обчислення арифметичних виразів і трансцендентних функцій. Використання команд співпроцесора ix87.»

Виконав: Перевірив:

Студент 2 курсу ФІОТ доц. Павлов В.Г.

Групи ІМ-13

Ситник Денис Олександрович

Номер у списку групи: 22

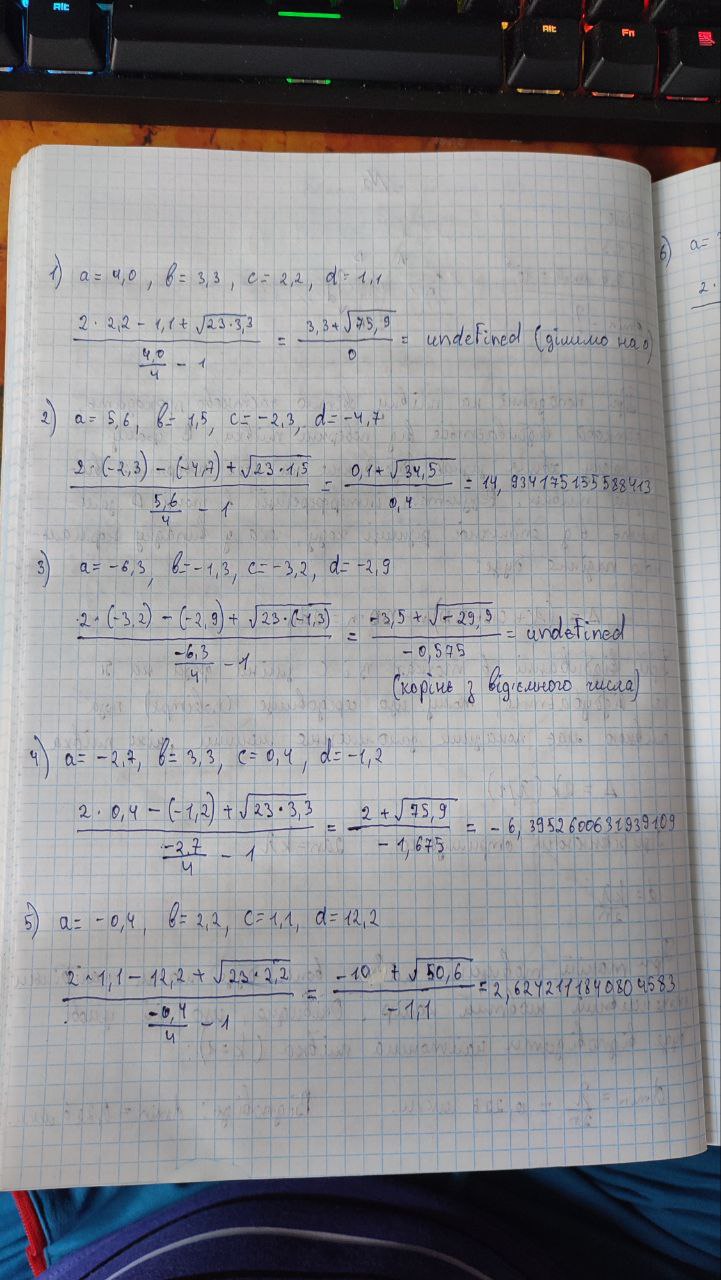
Київ – 2023

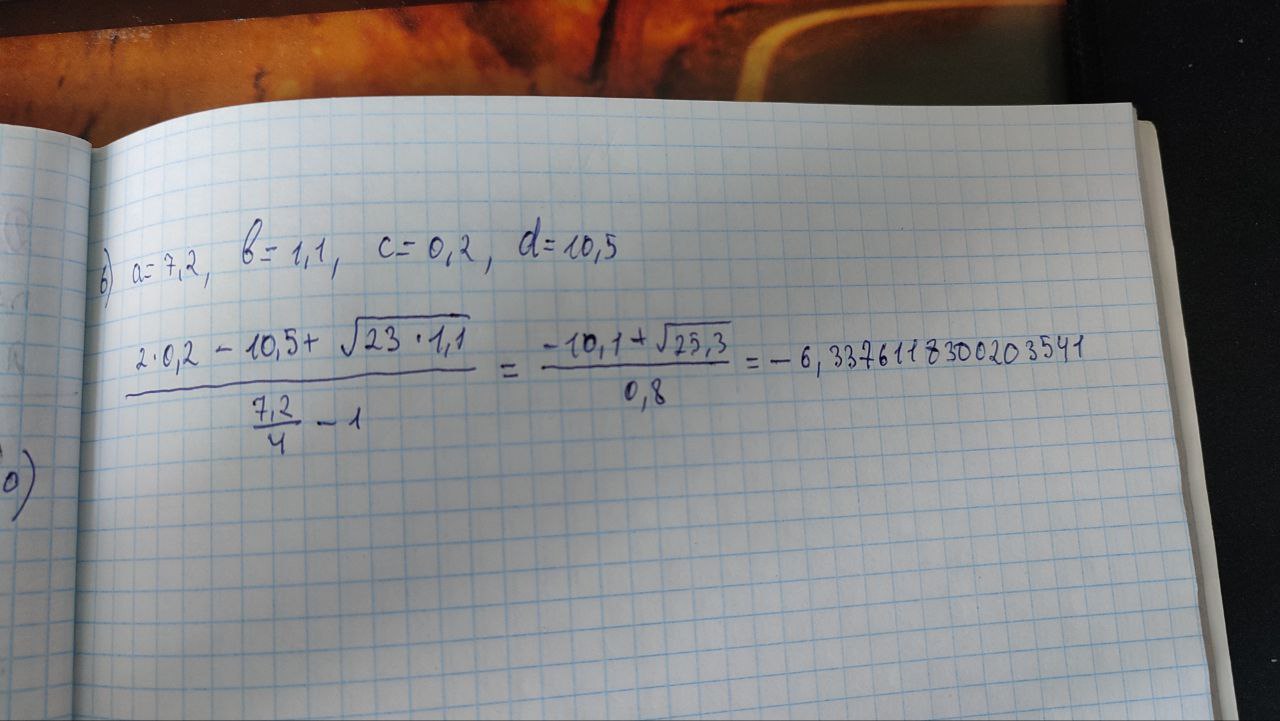
**Мета роботи:** вивчення команд Асемблера для арифметики з плаваючою комою і здобуття навичок виконання розрахунків з елементами масивів.

Номер у списку – 22, варіант 1:



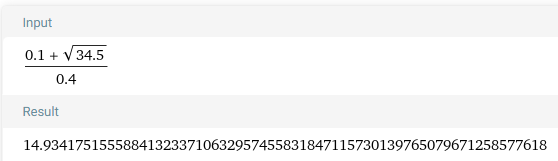
**Контрольні розрахунки:**

****

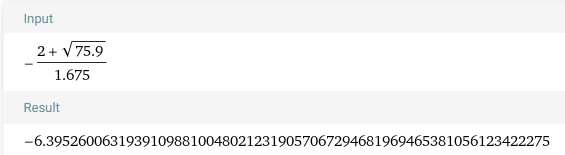
****

**Результати обчислень в інтернет-калькуляторі:**

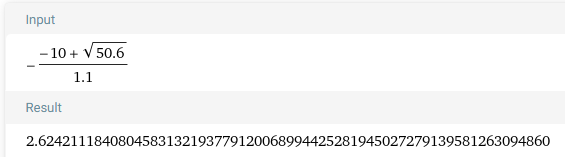
**2)**

****

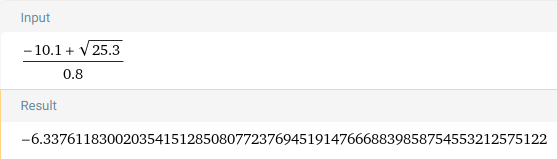
**4)**

****

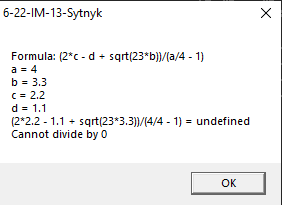
**5)**

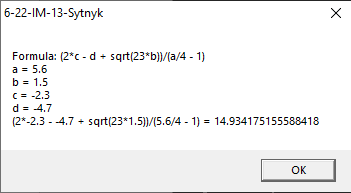
****

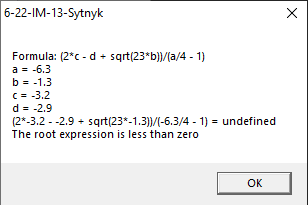
**6)**

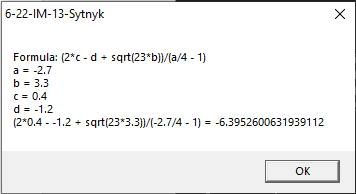
****

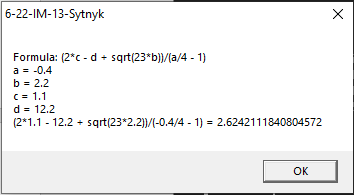
**Скріншоти виконання програми:**

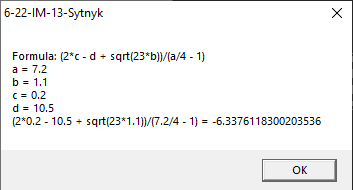
****

****

****

****

****

****

**Файл лістингу:**

coprocessorState macro

ftst

fnstsw ax

sahf

endm

successMessage macro

invoke wsprintf, addr buff, addr dataFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA, addr resultStr

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

errorMessage macro messageFormat

invoke wsprintf, addr buff, addr messageFormat,

addr valueA, addr valueB, addr valueC, addr valueD,

addr valueC, addr valueD, addr valueB, addr valueA

invoke szCatStr, addr data, addr buff

invoke MessageBox, 0, offset data, offset labTitle, 0

endm

00000000 .data?

00000000 00000020 [ numerator dt 32 dup(?) ;; long double

00000000000000000000

]

00000140 00000020 [ denominator dt 32 dup(?) ;; long double

00000000000000000000

]

00000280 00000020 [ result dq 32 dup(?)

0000000000000000

]

00000380 00000020 [ resultStr db 32 dup(?)

00

]

000003A0 00000020 [ valueA db 32 dup(?)

00

]

000003C0 00000020 [ valueB db 32 dup(?)

00

]

000003E0 00000020 [ valueC db 32 dup(?)

00

]

00000400 00000020 [ valueD db 32 dup(?)

00

]

00000420 00000020 [ data db 32 dup(?)

00

]

00000440 00000020 [ buff db 32 dup(?)

00

]

00000000 .data

00000000 36 2D 32 32 2D labTitle db "6-22-IM-13-Sytnyk", 0

49 4D 2D 31

33 2D 53 79

74 6E 79 6B

00

00000012 46 6F 72 6D 75 dataFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

6C 61 3A 20

28 32 2A 63

20 2D 20 64

20 2B 20 73

71 72 74 28

32 33 2A 62

29 29 2F 28

61 2F 34 20

2D 20 31 29

0A 61 20 3D

20 25 73 0A

62 20 3D 20

25 73 0A 63

20 3D 20 25

73 0A 64 20

3D 20 25 73

0A 28 32 2A

25 73 20 2D

20 25 73 20

2B 20 73 71

72 74 28 32

33 2A 25 73

29 29 2F 28

25 73 2F 34

20 2D 20 31

29 20 3D 20

25 73 0A 00

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = %s", 10, 0

00000083 46 6F 72 6D 75 zeroDenominatorFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

6C 61 3A 20

28 32 2A 63

20 2D 20 64

20 2B 20 73

71 72 74 28

32 33 2A 62

29 29 2F 28

61 2F 34 20

2D 20 31 29

0A 61 20 3D

20 25 73 0A

62 20 3D 20

25 73 0A 63

20 3D 20 25

73 0A 64 20

3D 20 25 73

0A 28 32 2A

25 73 20 2D

20 25 73 20

2B 20 73 71

72 74 28 32

33 2A 25 73

29 29 2F 28

25 73 2F 34

20 2D 20 31

29 20 3D 20

75 6E 64 65

66 69 6E 65

64 0A 43 61

6E 6E 6F 74

20 64 69 76

69 64 65 20

62 79 20 30

0A 00

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"Cannot divide by 0", 10, 0

0000010E 46 6F 72 6D 75 invalidDefinitionAreaFormat db "Formula: (2\*c - d + sqrt(23\*b))/(a/4 - 1)", 10,

6C 61 3A 20

28 32 2A 63

20 2D 20 64

20 2B 20 73

71 72 74 28

32 33 2A 62

29 29 2F 28

61 2F 34 20

2D 20 31 29

0A 61 20 3D

20 25 73 0A

62 20 3D 20

25 73 0A 63

20 3D 20 25

73 0A 64 20

3D 20 25 73

0A 28 32 2A

25 73 20 2D

20 25 73 20

2B 20 73 71

72 74 28 32

33 2A 25 73

29 29 2F 28

25 73 2F 34

20 2D 20 31

29 20 3D 20

75 6E 64 65

66 69 6E 65

64 0A 54 68

65 20 72 6F

6F 74 20 65

78 70 72 65

73 73 69 6F

6E 20 69 73

20 6C 65 73

73 20 74 68

61 6E 20 7A

65 72 6F 0A

00

"a = %s", 10,

"b = %s", 10,

"c = %s", 10,

"d = %s", 10,

"(2\*%s - %s + sqrt(23\*%s))/(%s/4 - 1) = undefined", 10,

"The root expression is less than zero", 10, 0

000001AC valuesA dq 4.0, 5.6, -6.3, -2.7, -0.4, 7.2

4010000000000000

4016666666666666

C019333333333333

C00599999999999A

BFD999999999999A

401CCCCCCCCCCCCD

000001DC valuesB dq 3.3, 1.5, -1.3, 3.3, 2.2, 1.1

400A666666666666

3FF8000000000000

BFF4CCCCCCCCCCCD

400A666666666666

400199999999999A

3FF199999999999A

0000020C valuesC dq 2.2, -2.3, -3.2, 0.4, 1.1, 0.2

400199999999999A

C002666666666666

C00999999999999A

3FD999999999999A

3FF199999999999A

3FC999999999999A

0000023C valuesD dq 1.1, -4.7, -2.9, -1.2, 12.2, 10.5

3FF199999999999A

C012CCCCCCCCCCCD

C007333333333333

BFF3333333333333

4028666666666666

4025000000000000

0000026C one dq 1.0

3FF0000000000000

00000274 two dq 2.0

4000000000000000

0000027C four dq 4.0

4010000000000000

00000284 twentyThree dq 23.0

4037000000000000

00000000 .code

00000000 main:

00000000 BD 00000000 mov ebp, 0

.while ebp < 6

;; Converting numbers to strings

invoke FloatToStr2, valuesA[ebp \* 8], addr valueA

invoke FloatToStr2, valuesB[ebp \* 8], addr valueB

invoke FloatToStr2, valuesC[ebp \* 8], addr valueC

invoke FloatToStr2, valuesD[ebp \* 8], addr valueD

0000006A 9B DB E3 finit

;; Calculating denominator

0000006D DD 05 0000027C R fld four ;; 4 in ctack

00000073 DD 04 ED fld valuesA[ebp \* 8] ;; a in ctack

000001AC R

0000007A D8 F1 fdiv st(0), st(1) ;; a/4

0000007C DD 05 0000026C R fld one ;; 1 in stack

00000082 D9 C9 fxch st(1) ;; change positions

00000084 D8 E1 fsub st(0), st(1) ;; a/4 - 1

coprocessorState

00000086 D9 E4 1 ftst

00000088 DF E0 1 fnstsw ax

0000008A 9E 1 sahf

0000008B 0F 84 000000CF jz zeroDenominator

00000091 DB 3D 00000140 R fstp denominator ;; saving denominator

;; Calculating numerator

00000097 DD 05 00000284 R fld twentyThree ;; 23 in stack

0000009D DC 0C ED fmul valuesB[ebp \* 8] ;; 23\*b

000001DC R

000000A4 D9 FA fsqrt ;; sqrt(23\*b)

coprocessorState

000000A6 D9 E4 1 ftst

000000A8 DF E0 1 fnstsw ax

000000AA 9E 1 sahf

000000AB F6 C4 40 test ah, 01000000b

000000AE 0F 85 0000010A jnz invalidDefinitionArea

000000B4 DD 05 00000274 R fld two

000000BA DC 0C ED fmul valuesC[ebp \* 8]

0000020C R

000000C1 DC 24 ED fsub valuesD[ebp \* 8]

0000023C R

000000C8 D8 C1 fadd st(0), st(1)

000000CA DB 3D 00000000 R fstp numerator ;; saving numerator

;; Preparing for division

000000D0 DB 2D 00000000 R fld tbyte ptr [numerator] ;; long double

000000D6 DB 2D 00000140 R fld tbyte ptr [denominator] ;; long double

000000DC DE F9 fdivp st(1), st(0) ;; numerator/denominator

000000DE DD 1D 00000280 R fstp qword ptr [result] ;; saving result in double format

invoke FloatToStr2, result, addr resultStr ;; converting result into string

successMessage

0000015B E9 000000BC jmp next

00000160 zeroDenominator:

errorMessage zeroDenominatorFormat

000001BC EB 5E jmp next

000001BE invalidDefinitionArea:

errorMessage invalidDefinitionAreaFormat

0000021A EB 00 jmp next

0000021C next:

0000021C 45 inc ebp

0000021D C6 05 00000420 R mov data, 0h

00

.endw

invoke ExitProcess, 0

end main

**Висновок:**

Виконавши лабораторну роботу я вивчив команди Асемблера для арифметики з плаваючою комою і здобув навички виконання розрахунків з елементами масивів, навчився працювати із стеком та рахувати трансцендентні функції.  
Я розробив програму що рахує значення по заданій формулі. Вихідні дані задав у форматі double (масиви з числами), проміжні результати у форматі long double (numerator і denominator), та остаточні результати знову у форматі double (result). Підставив різні аргументи, такі, що формують усі можливі варіанти знаків у чисельнику і знаменнику та викликають помилку в обрахунках (ділення на нуль або корінь з від’ємного числа), отримав результати. Порівнявши їх із контрольними розрахунками, можна дійти до висновку, що отримані результати є достовірними.

Слід зауважити, що результати контрольних розрахунків та результати обчислення компілятора відрізняються в останніх цифрах (є незначна похибка).