**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 № 8**

з дисципліни

«Системне програмування»

на тему

“Розробка і використання динамічних бібліотек”

Виконав:

Перевірив:

студент групи ІП-93 доцент

Марченко Максим Олегович Павлов В.Г.

номер залікової книжки: 9316

Варіант 13

Київ 2021

Мета роботи : Вивчення прийомів розробки і використання процедур, представлених у вигляді динамічних бібліотек.

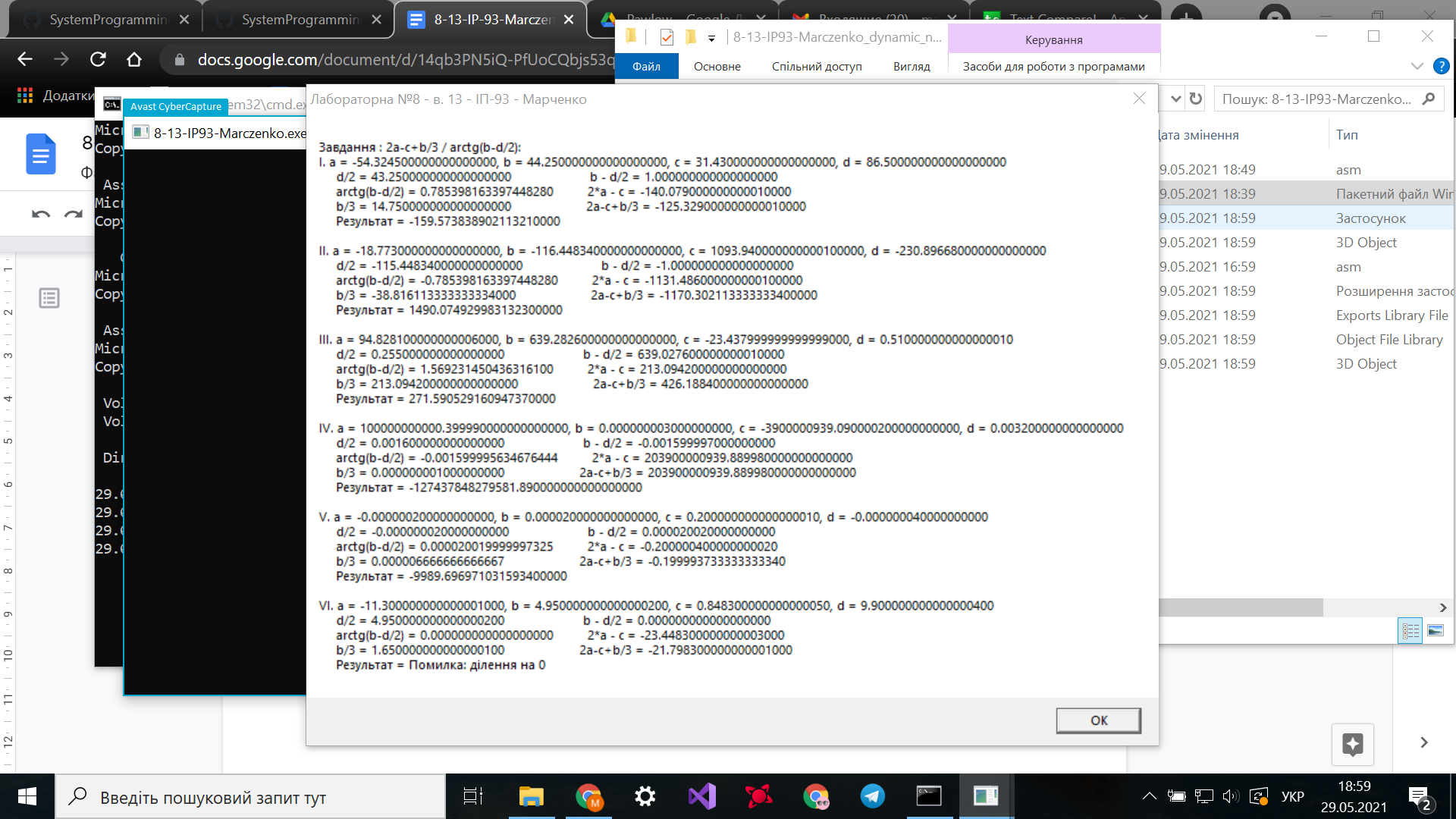
DLL ( Dynamic Link Library) - підтип загальних бібліотек у Windows. Вони використовуються тоді, коли різні файли-компоненти програми використовують одну й ту саму функцію. Тоді, таку функцію записують в окремий файл, який і стає бібліотекою, а його підключають до кожного компоненту, що вимагає певної функціональності.

Застосунок може використовувати DLL двома способами:

1. Статично - під’єднати бібліотеку на етапі початкового завантаження програмного файлу. Для цього, у файлі запуску потрібно описати прототип функцій, що будуть запускатися
2. Динамічно - додаток сам завантажує і вивантажує бібліотеку за допомогою системних команд LoadLibrary, GetProcAddress ,FreeLibrary. Тоді, для виклику певної процедури використовується її адреса, отримана при виконанні GetProcAddress у регістрі еах. Аналогічно, адреса бібліотеки отримується при виконанні LoadLibrary.

Динамічне підключення є більш ефективним, і менш схильним до помилок, оскільки програма сама контролює, коли потрібно завантажувати і вивантажувати бібліотеку, а також які саме функції з неї запускати. При статичному підключенні все це повинен прописати користувач у програмному та .def файлах.

Скріншот виконання програми



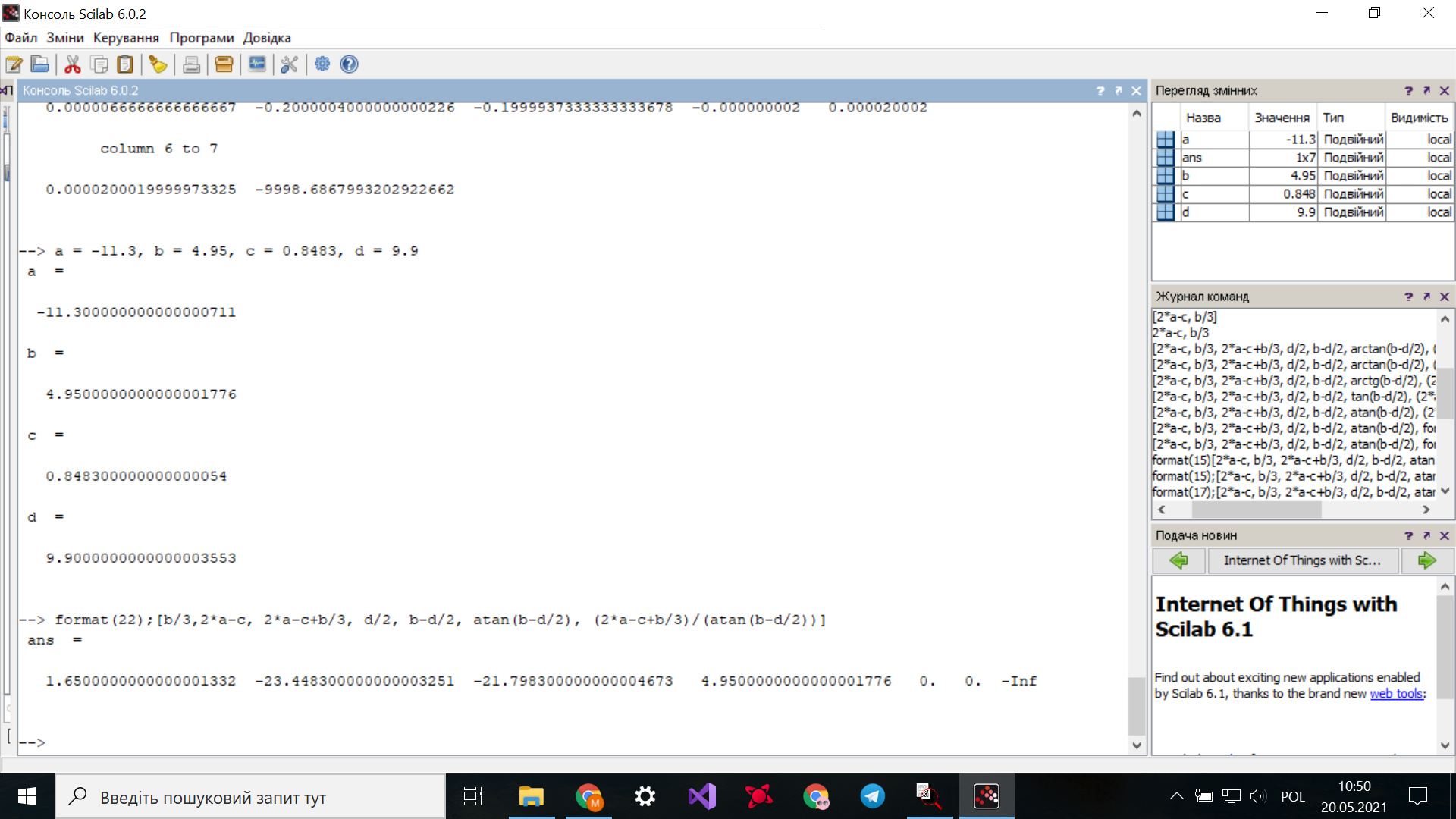
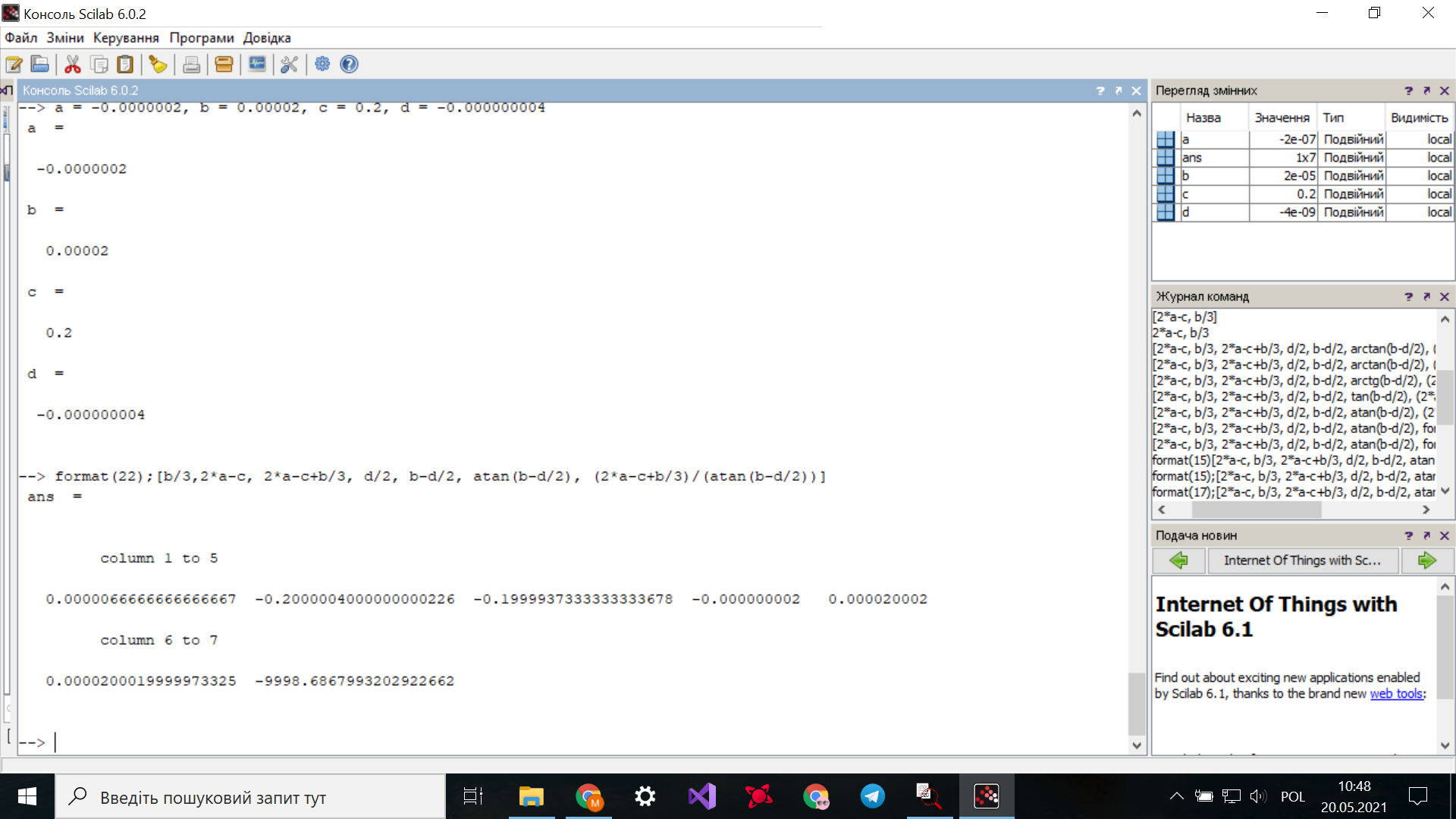
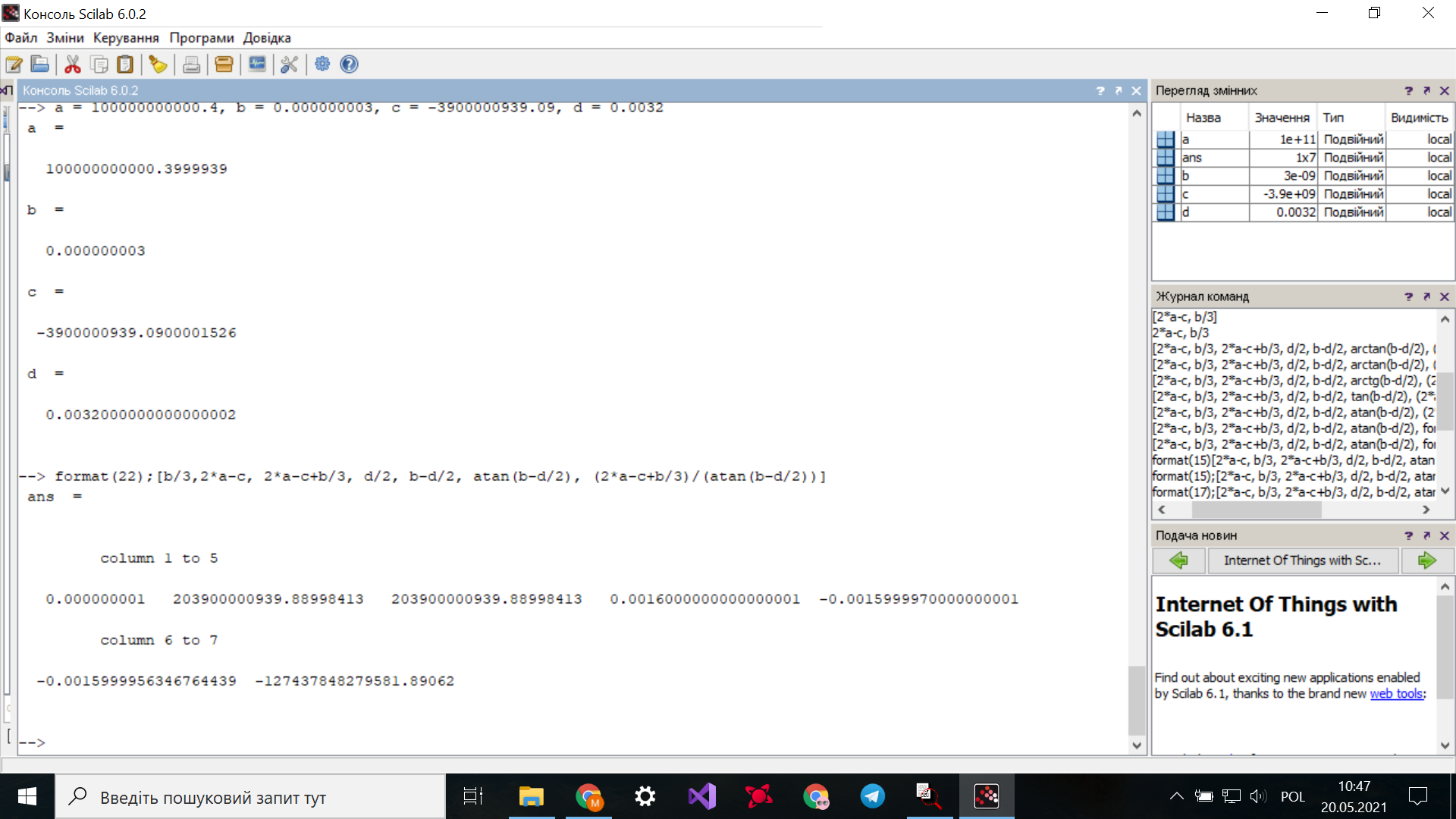
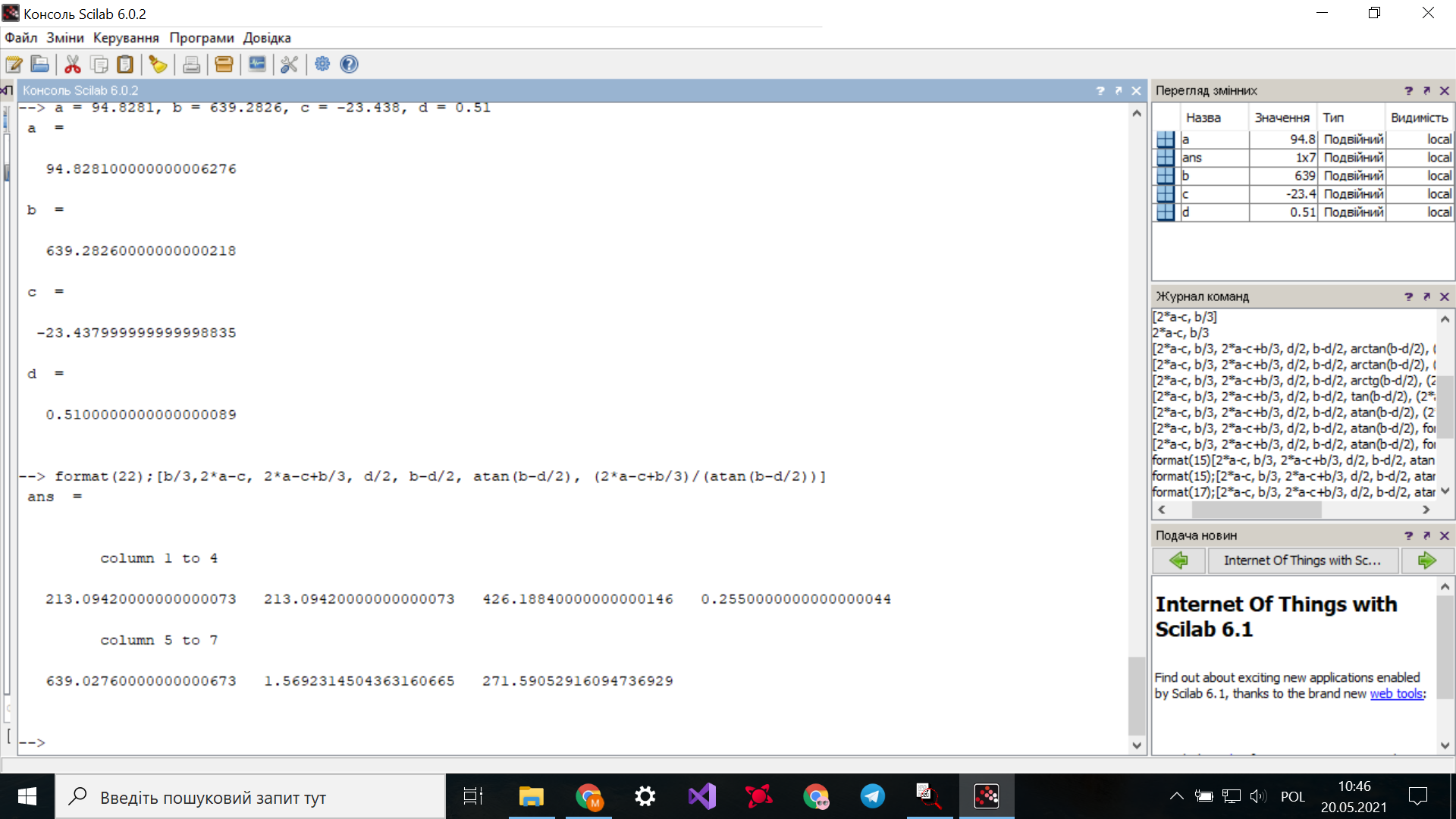
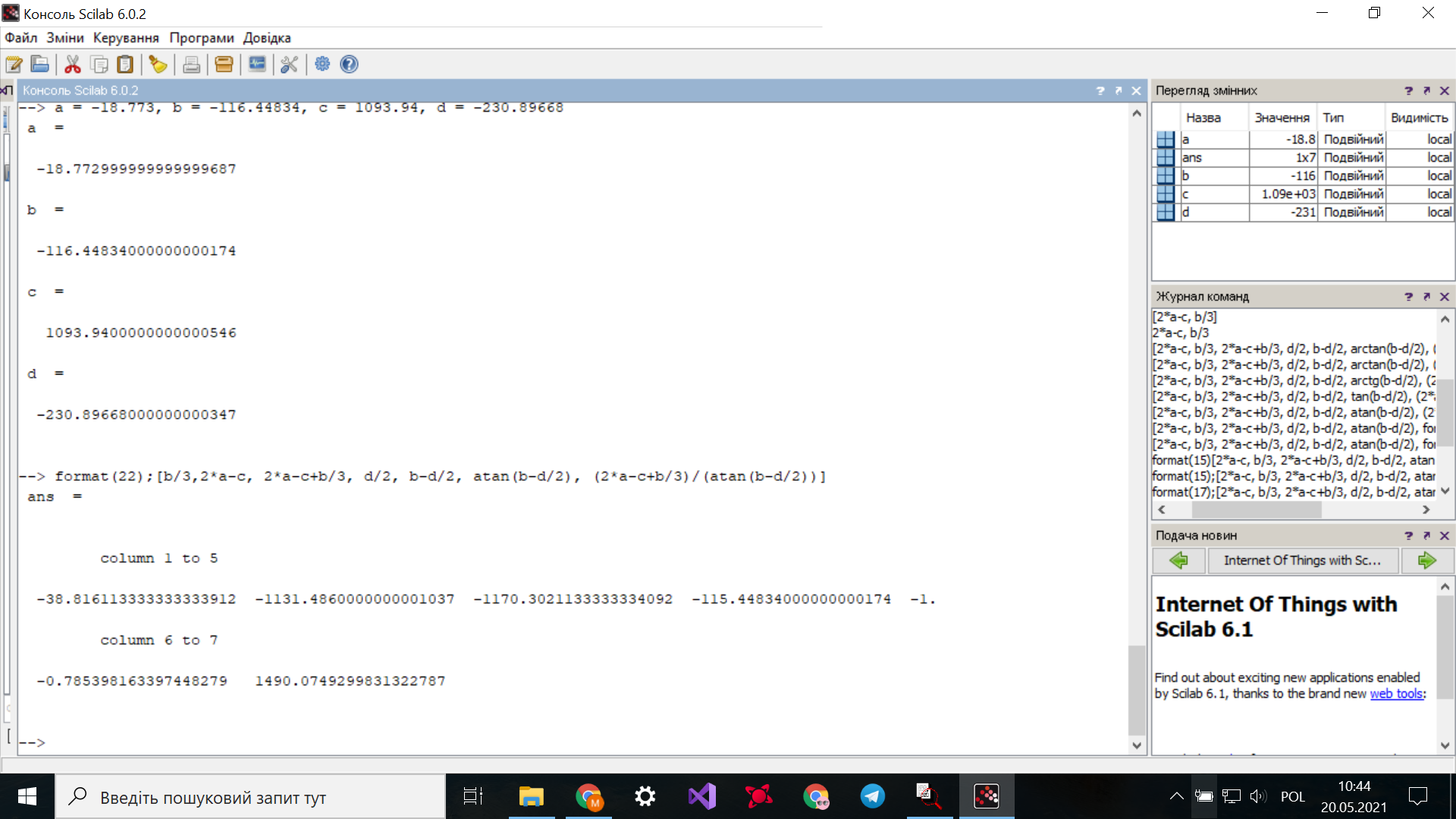
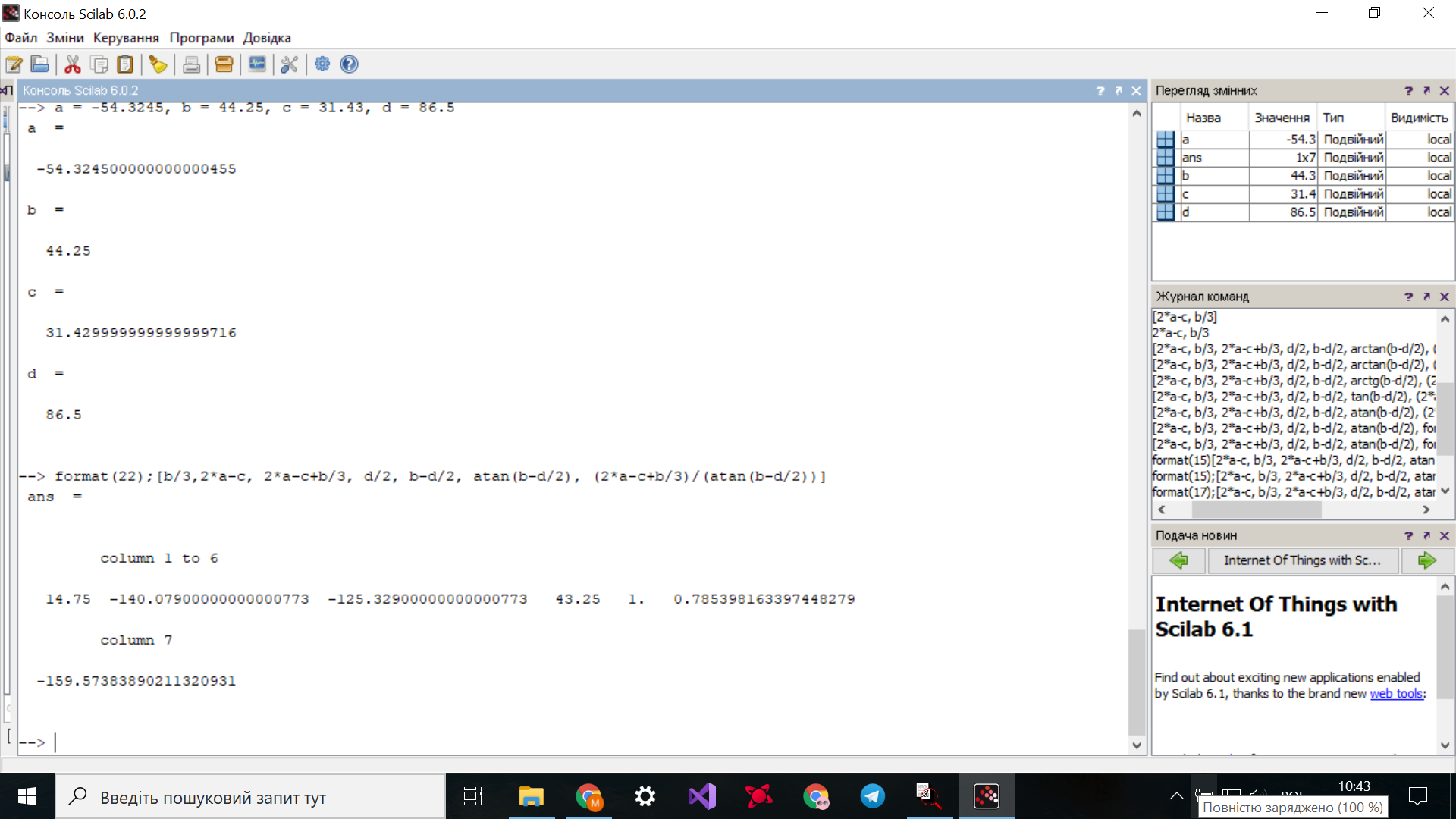
На скріншоті зображено виконання програми з 6 наборами вхідних даних, 5 з яких - коректні, 1 - спричинює ділення на 0.

Для знаходження помилки ділення на 0 знаменник виразу порівнюється з нулем перед фінальною дією ділення чисельник на знаменник. Якщо знаменник рівний нулю, то ця дія не виконується, замість її результату виводиться повідомлення про помилку.

Контрольні розрахунки

Розрахунки проводитимуться в тому ж порядку, як і в програмі. Черговість дій:

1. b/3
2. 2a-c
3. 2a-c+b/3
4. d/2
5. b-d/2
6. arctg(b-d/2)
7. (2a-c+b/3) / arctg(b-d/2)



Дії для контрольних розрахунків проведені в тій ж послідовності, що й в програмі.

Як видно, результат виконання програми програми та контрольні розрахунки збігаються з точністю до заокруглення для всіх вхідних даних, включаючи випадки помилок, що підтверджує правильність програми. Похибка у 5) прикладі пояснюється тим, що у ньому використовуються дуже малі числа, і тому будь-яка похибка заокруглення результату при переведенні результату з двійкового в десятковий режим (що спостерігається як у масм, так і в прикладах перевірки) створює значний вплив на підрахунок фінального результату.

Висновки

При виконанні роботи я навчився створювати та використовувати динамічні бібліотеки у своїх програмах, що забезпечує її модульність, тобто полегшує можливість використання певних процедур в інших проєктах. Є два методи підключення бібліотек до програми: статичний та динамічний. Динамічне підключення є більш ефективним, і менш схильним до помилок, оскільки програма сама контролює, коли потрібно завантажувати і вивантажувати бібліотеку, а також які саме функції з неї запускати. При статичному підключенні все це повинен прописати користувач у програмному та .def файлах.

Додаток: Код програми 8-13-IP-93-Marczenko (статичний)

.486

.model flat, stdcall

option casemap :none

INCLUDE \masm32\include\user32.inc

INCLUDE \masm32\include\kernel32.inc

INCLUDE \masm32\include\msvcrt.inc

INCLUDELIB \masm32\lib\kernel32.lib ; for ExitProcess

INCLUDELIB \masm32\lib\user32.lib ; for MessageBox

INCLUDELIB \masm32\lib\msvcrt.lib ; for crt\_sprintf

INCLUDELIB 8-13-IP93-Marczenko-lib.lib ; for enumerate

enumerate PROTO :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD, :PTR QWORD

calc MACRO i, buff

local cont

local no\_error

invoke enumerate, addr a[i\*8], addr b[i\*8], addr cc[i\*8], addr d[i\*8], addr res1[i\*8], addr res2[i\*8], addr res3[i\*8], addr res4[i\*8], addr res5[i\*8], addr res6[i\*8], addr results[i\*8]

cmp eax, 1

jne no\_error

invoke crt\_sprintf, addr buffer, addr error\_message

jmp cont

no\_error:

invoke crt\_sprintf, addr buffer, addr number, results[i\*8]

cont:

invoke crt\_sprintf, addr buffer1, addr number, res1[i\*8]

invoke crt\_sprintf, addr buffer2, addr number, res2[i\*8]

invoke crt\_sprintf, addr buffer3, addr number, res3[i\*8]

invoke crt\_sprintf, addr buffer4, addr number, res4[i\*8]

invoke crt\_sprintf, addr buffer5, addr number, res5[i\*8]

invoke crt\_sprintf, addr buffer6, addr number, res6[i\*8]

invoke crt\_sprintf, buff, addr out\_message, a[i\*8], b[i\*8], cc[i\*8], d[i\*8],addr buffer1,addr buffer2,addr buffer3,addr buffer4,addr buffer5,addr buffer6, addr buffer

ENDM

.data

header DB "Лабораторна №8 - в. 13 - ІП-93 - Марченко", 0

template DB "Завдання : 2a-c+b/3 / arctg(b-d/2):", 10,

"I. %s", 10,

"II. %s", 10,

"III. %s", 10,

"IV. %s", 10,

"V. %s", 10,

"VI. %s", 0

out\_message DB "a = %.18f, b = %.18f, c = %.18f, d = %.18f",10,

" d/2 = %s"," b - d/2 = %s", 10,

" arctg(b-d/2) = %s"," 2\*a - c = %s", 10,

" b/3 = %s"," 2a-c+b/3 = %s", 10, " Результат = %s",10,0

number DB "%.18f", 0

error\_message DB "Помилка: ділення на 0", 0

current\_buff DD 0

out\_buffer DB 4096 DUP (0)

buff\_ex1 DB 512 DUP (0)

buff\_ex2 DB 512 DUP (0)

buff\_ex3 DB 512 DUP (0)

buff\_ex4 DB 512 DUP (0)

buff\_ex5 DB 512 DUP (0)

buff\_ex6 DB 512 DUP (0)

buffersize = $ - buff\_ex6

buffer DB 256 DUP (0)

buffer1 DB 64 DUP (0)

buffer2 DB 64 DUP (0)

buffer3 DB 64 DUP (0)

buffer4 DB 64 DUP (0)

buffer5 DB 64 DUP (0)

buffer6 DB 64 DUP (0)

results DQ 6 DUP (0)

res1 DQ 6 DUP (0)

res2 DQ 6 DUP (0)

res3 DQ 6 DUP (0)

res4 DQ 6 DUP (0)

res5 DQ 6 DUP (0)

res6 DQ 6 DUP (0)

a DQ -54.3245, -18.773, 94.8281, 100000000000.4, -0.0000002, -11.3

b DQ 44.25, -116.44834, 639.2826 , 0.000000003 , 0.00002, 4.95

cc DQ 31.43, 1093.94, -23.438, -3900000939.09 , 0.2, 0.8483

d DQ 86.5, -230.89668, 0.51 , 0.0032 , -0.00000004, 9.9

MB\_OK EQU 0

.code

start:

mov esi, 0

mov current\_buff, offset buff\_ex1

cycle1:

calc esi, current\_buff

ADD current\_buff, buffersize

INC esi

CMP esi, 6

JB cycle1

invoke crt\_sprintf, addr out\_buffer, addr template,

addr buff\_ex1, addr buff\_ex2,

addr buff\_ex3, addr buff\_ex4,

addr buff\_ex5, addr buff\_ex6

invoke MessageBox, 0, addr out\_buffer, addr header, MB\_OK

invoke ExitProcess, 0

END start