НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Кафедра прикладної математики

РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА

з кредитного модуля

"Програмування 1. Основи програмування"

на тему:

«Програма розв’язання системи ? нелінійних рівнянь графічним методом»

Виконала Карачун Анастасія Ігорівна

група КМ-02 факультет прикладної математики

N залікової книжки КМ-0209

 Керівник  Олефір О.С. (                              )

                                                                              "14" грудня 2020р.

Захищена з оцінкою\_\_\_\_\_\_\_\_\_\_\_

Київ 2020

ЗМІСТ

[1 ПОСТАНОВКА ЗАДАЧІ 3](#_heading=h.30j0zll)

[1.1](#_heading=h.1fob9te) Дослідження проблеми 3

[2 ПРОЕКТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ 5](#_heading=h.3znysh7)

[2.1. Методи вирішення задачі 5](#_heading=h.2et92p0)

[2.2. Проектування алгоритмів 6](#_heading=h.tyjcwt)

[2.2.1](#_heading=h.3dy6vkm) Структура програмного забезпечення 7

[2.2.2](#_heading=h.1t3h5sf) Опис розроблених алгоритмів 8

[2.2.3](#_heading=h.4d34og8) Засоби керування програмами 8

[3 РЕАЛІЗАЦІЯ ПРОГРАМИ 9](#_heading=h.2s8eyo1)

[3.1. Опис вхідних даних 9](#_heading=h.17dp8vu)

[3.2. Опис результатів 9](#_heading=h.3rdcrjn)

[3.3. Контрольні приклади 9](#_heading=h.26in1rg)

[3.4. Експериментальні розрахунки 10](#_heading=h.lnxbz9)

[ВИСНОВКИ 11](#_heading=h.35nkun2)

[ЛІТЕРАТУРА 12](#_heading=h.1ksv4uv)

[ДОДАТОК А 13](#_heading=h.44sinio)

[ДОДАТОК Б 16](#_heading=h.2jxsxqh)

[ДОДАТОК В 19](#_heading=h.z337ya)

# 1 ПОСТАНОВКА ЗАДАЧІ

* 1. Дослідження проблеми

Розв’язання систем нелінійних рівнянь – одна із складних задач обчислювальної математики. Складність полягає у тому, щоб з’ясувати: чи має система розв’язок, і, якщо – так, то скільки.

До них зводиться багато практичних задач, наприклад, розрахунки нелінійних електричних кіл та систем керування, розв’язання нелінійних диференціальних рівнянь, аналіз стійкості систем шляхом оцінки їх власних значень та ін.

Якщо для найпростіших видів алгебраїчних рівнянь (не вище третього степеня) існують точні аналітичні формули, то для трансцендентних рівнянь (тих, що містять тригонометричні, експоненціальні, логарифмічні або інші спеціальні функції) і будь-яких систем рівнянь таких методів взагалі не існує і слід користуватися тільки наближеними ітераційними методами та алгоритмами, а також використовуючи графічний метод.

Графічний метод особливо ефективний при якісному аналізі рівняння, коли потрібно визначити, чи існує корінь взагалі, або число можливих коренів.

Загалом, мету роботи можна розділити на декілька підпунктів:

* Вивчення графічного методу розв’язку системи нелінійних рівнянь;
* Отримання базових знань мови програмування Python, необхідних для реалізації завдання;
* Створення схеми алгоритму дій програми та її взаємодії з користувачем;
* Написання допоміжних функцій, що будуть потрібними для виконання програмою поставленої задачі;
* Забезпечення легкості та зрозумілості використання програми, щоб скористатись нею міг навіть користувач, не знайомий із програмуванням;
* Передбачення коректного вводу та виводу даних з метою отримання чітких та правильних результатів та недопуску помилок в ході виконання програми;
* Створення ескізів інтерфейсу програми (за можливості – його реалізація)

Для використання програми необхідним є встановлення на робочому пристрої користувача компілятора мови програмування Python (версії 3.6+) та бібліотек NumPy та Matplotlib, котрі використовуються у програмі для полегшення проміжних завдань та виведення графіків заданих рівнянь на екран.

# 2 ПРОЕКТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ

## 2.1. Методи вирішення задачі

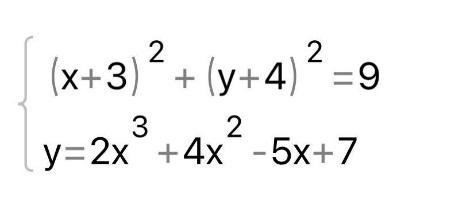
Геометричний сенс розв’язку системи рівняння  як алгебраїчного, так і трансцендентного, полягає у знаходженні точки перетину графіків функцій.

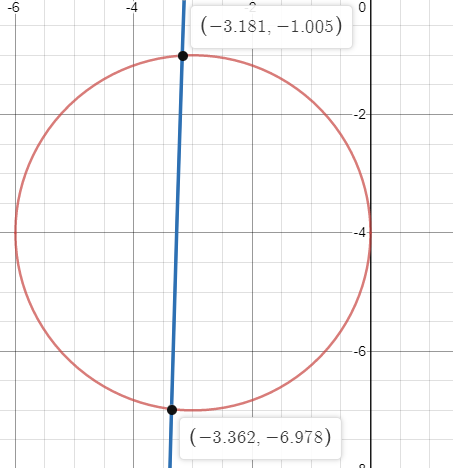
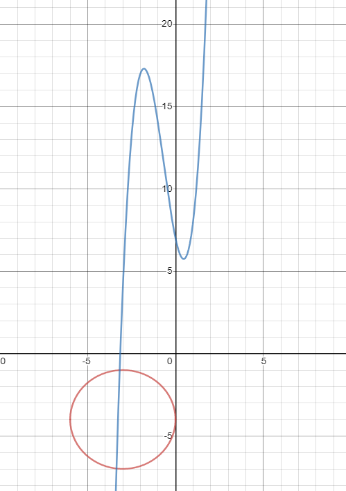
Графічний метод зводиться до побудови [графіків функцій](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D1%96%D0%BA_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%97) та візуального визначення точки, де вона перетинає [вісь ординат](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D1%81%D1%8C_%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82). Іноді побудова графіків функцій  та   складна, але рівняння можна переписати у вигляді ,,

де, – функції з простими графіками. Тоді графічний метод зводиться до знаходження точки перетину двох функцій.   
 Результат треба одержати чисельний, використовується покрокове  
зменшення інтервалу побудови графіків поблизу точки перетину функцій.

Приклад методу вирішення задачі:

Задана система нелінійних рівнянь:



Побудуємо графіки обох рівнянь на одній координатній площині. З зображення видно, що є дві точки перетину. 

Щоб знайти ці точки перетину здійснюємо координатне наближення до місць перетину та дізнаємося точні значення.

## 2.2. Проектування алгоритмів

Запуск програми являє собою виведення на екран початкового діалогового вікна, а також інформаційних повідомлень про склад і структуру програми, та вимоги до формату даних.

На етапі введення користувачем даних, використовується бібліотека NumPy, аби мати можливість задавати математичні формули (sin, arccos і тд). Користувач має вводити рівняння вже зведене до залежності y від x. Перевіряється наявність такого виду рівняння у бібліотеках для подальшої побудови, також визначається коректність вводу значень. У разі виникнення помилки роботи з форматом або ігнорування вимог до введення програма на екран виводиться оповіщення про помилку.

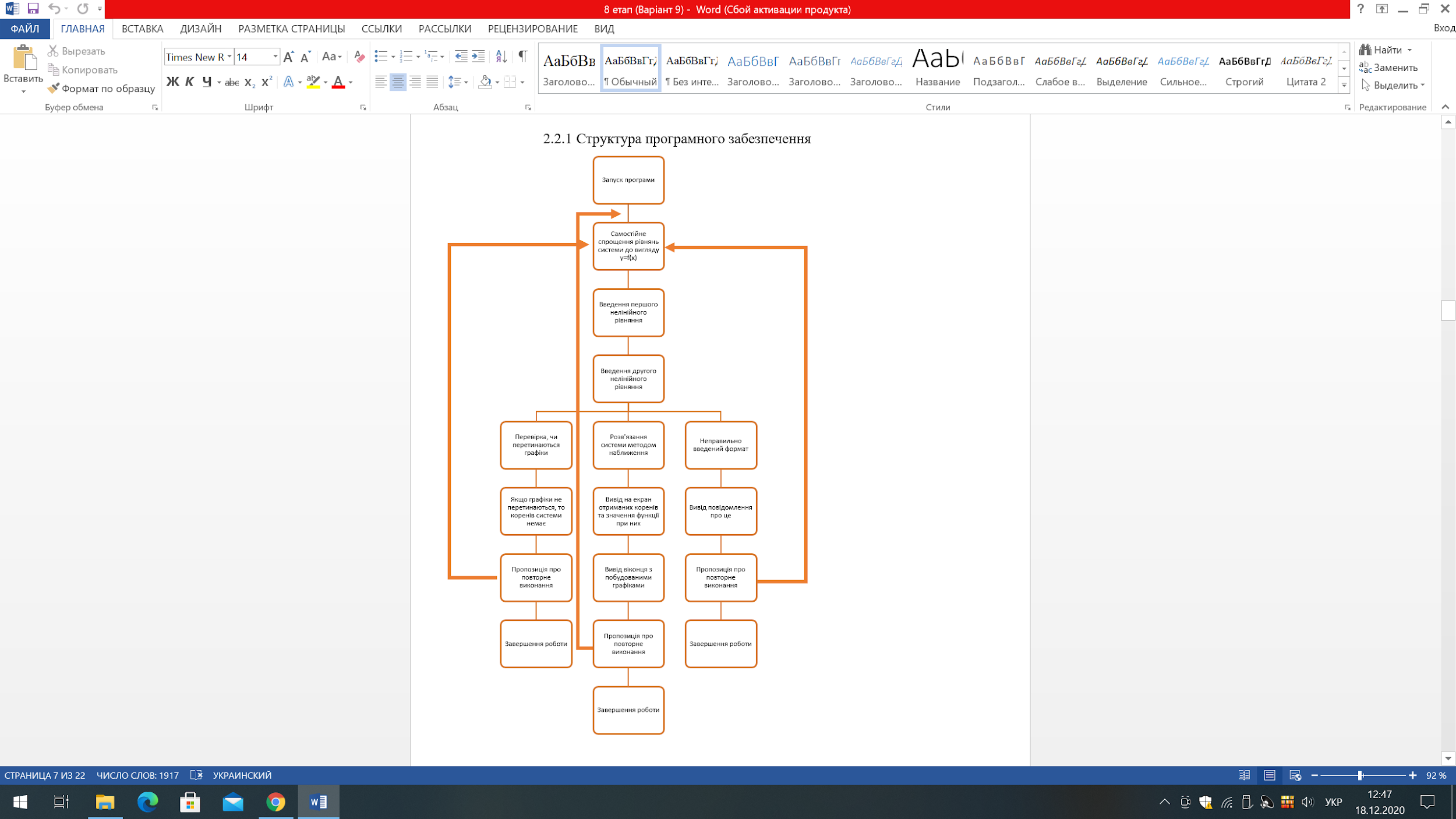
Однак перед тим, як повністю завершити свою роботу, на екран буде виведено питання, чи хоче користувач продовжити роботу. Це відбувається за допомогою цикла while. Щоб продовжити роботу потрібно ввести ‘yes’.

Якщо етап введення даних пройдено успішно, програма починає аналізувати рівняння системи, а також їхні графіки. Методом наближення до точок перетину графіків, знаходимо їхні координати, використовуючи метод abs до операції пошуку відстані між побудованими графіками. Коли ця відстань дорівнюватиме числу меншому за 0.1, виводиться ордината кореня х, при якому відбувається перетин. Усе супроводжується відповідними записами на екрані.

Насамкінець, програма виводить корені рівняння, якщо вони знайдені, а також координатну площину з виведеними за допомогою бібліотеки Matplotlib графіками нелінійних рівнянь системи.

Програма завершує свою роботу, коли користувач на запит продовження дій, вводить будь-який інший запис, окрім ‘yes’.

### Структура програмного забезпечення



### Опис розроблених алгоритмів

Повний текст програми подано у Додатку А. Детальний хід дій можна побачити в блок-схемі в Додатку Б (Блок-схема 1).

Отже, детальніше про деякі підпрограми і функції. У розрахунково-графічній роботі використано бібліотеки NumPy та Matplotlib для прийняття формату рівнянь та створення графічного інтерфейсу (вікна, у якому виводяться графіки системи). Наприклад, модуль np.linspace використовується для задання масиву можливих коренів системи на інтервалі (-10, 10). Окремо задано функцію check, що перевіряє рівняння системи на правильність. Використовуючи цикл for та умовну конструкцію if, ми проходимо по абсолютним значенням рівняння та шукаємо місця, де відстань між двома графіками менша за 0,1. Також задана змінна k, яка на початку дорівнює нулю, а з кожним знайденим коренем збільшується на 1. За умови, що k=0, система не має розв’язків.

Далі за допомогою модуля plt, вказуємо параметри для створення вікна з графічним інтерфейсом та викликаємо, використовуючи plt.show(). У разі виникнення помилок вводу, функція виводить на екран повідомлення про помилку.

Насамкінець, можна продовжити роботу в програмі, написавши yes, або вийти з неї, ввівши будь-який інший символ. У другому випадку, буде виведено на екран текст з повідомленням про завершення роботи.

### Засоби керування програмами

Можливостями користувача керувати програмою є такі дії:

1. Самостійне спрощення рівнянь системи;
2. Ввід даних (в цьому випадку систему двох нелінійних рівнянь);
3. Можливість декілька разів повторити роботу з функцією.

Робота програми може виконуватися та контролюватися з командного рядка. Однак також можливий варіант оформлення окремого вікна для розв’язання нелінійних рівнянь графічним методом. Приклад інтерфейсу поданий у Додатку Б (рис.2).

# 3 РЕАЛІЗАЦІЯ ПРОГРАМИ

## 3.1. Опис вхідних даних

На вхід програма приймає нелінійні рівняння, в яких вже виділено залежність у від х. Так наприклад, якщо є рівняння: , то користувач має вводити праву частину цієї функції: , інакше – програма видаватиме помилку. Основною вимогою є правильність введення рівняння, щоб у правій частині була лише одна змінна. Тільки після введення двох рівнянь програма виводить або помилку, або результати обрахунків.

## 3.2. Опис результатів

По-перше, як результат розрахунків програма виводить на екран отримані округлені до десятих корені рівняння з відповідними значеннями функції в цій точці. Коли система заданих нелінійних рівнянь немає коренів, результатом роботи програми буде відповідне повідомлення.

По-друге, на екрані з’являється віконечко – координатна площина, у якій зображені графіки рівнянь системи та їхній перетин. Виведені результати у випадках, коли система немає коренів, має корінь та неправильно введена, можна побачити на рисунку 1 в пунктах а, б і в відповідно (див. Додаток В рис.1). Також програма пропонує спробувати розрахунок ще раз.

## 3.3. Контрольні приклади

Для того, щоб цілковито бути впевненими в роботі програми, особисто виконано контрольні приклади та отримано результати.

Приклад 1

На одній координатній площині побудовано графіки обох рівнянь (див. Додаток В рис. 2(а)). На даному графіку червоним кольором зазначено перше рівняння (тригонометричне), а синім – друге (квадратне). Лише подивившись на зображення, можна побачити, що графіки цієї системи перетинаються в двох точках, тож вона має два розв’язки. Продовжуючи розв’язання, збільшено масштаб графіка для детального визначення розв’язків (див. Додаток В рис. 2(б)).

Отже, знайдено корені системи нелінійних рівнянь: х = - 0,2 та х = 1,3.

Приклад 2

Другий приклад розглянуто за такою ж схемою, як і перший. На площині побудовано графіки обох рівнянь системи (див. Додаток В рис. 3(а)). На даному графіку червоним кольором позначено перше рівняння (функція квадратного кореня), а синім – друге (тригонометричне). Графіки цієї системи мають безліч розв’язків, але тільки два таких, що належать інтервалу (-10;10). Після збільшення масштабу графіка, знайдено значення: х = -1,1 та х = 2,4 (див. Додаток В рис. 3(б)).

## 3.4. Експериментальні розрахунки

Програму протестовано на контрольних прикладах, наведених вище. На графіках є позначки, яким кольором задане кожне з рівнянь. Усі значення при розрахунках збігаються, що доводить точність та правильність роботи створеної програми. Для того, щоб переконатися у цьому, варто переглянути скріншоти тестування на першому та другому контрольних прикладах (див. Додаток В рис.4(а, б))

# ВИСНОВКИ

У результаті виконання розрахунково-графічної роботи була досліджена тема розв’язання системи нелінійних рівнянь графічним методом. Відповідно до плану виконання розрахункової роботи, був детально вивчений метод вирішення поставленої задачі, обрахунок контрольних прикладів, спроектована архітектура майбутньої програми та можливий варіант користувацького інтерфейсу. Сам алгоритм програми написаний мовою Python. Наприкінці був оформлений звіт та пояснювальна записка. Щодо кожного виконаного етапу відбувалося звітування відповідно до зазначених дат.

Отже, було набуто навички розробки, проектування та тестування програмного забезпечення мовою Python, а також вміння застосування їх при вирішенні прикладних задач. Відбулося ознайомлення з повним циклом алгоритмізації рішення прикладної задачі – від аналізу та дослідження заданої проблеми і до експериментальних розрахунків та оформлення пояснювальної записки.

# ЛІТЕРАТУРА

1. Квєрний Р.Н. Комп’ютерне моделювання систем та процесів Част.1 / Квєрний Р.Н., Богач І.В., Бойко О.Р. / [Електронний ресурс] -

<https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp'yuterne_modelyuvannya_system_procesiv/t1/3..htm>

1. Онлайн-калькулятор для побудови графіків / [Електронний ресурс] -

<http://matematikam.ru/calculate-online/grafik.php>

# ДОДАТОК А

import matplotlib.pyplot as plt

from numpy import sin, cos, tan, arcsin, arccos, arctan

import numpy as np

intoduction ='''

РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА з дисципліни: «Програмування» 1-й семестр

На тему: "Програма розв’язання системи нелінійних рівнянь графічним методом"

Варіант 9

Керівник: Олефір О.С.

Виконала студентка групи КМ-02: Карачун Анастасія'''

description = '''

Програма призначена розв’язувати системи нелінійних рівнянь графічним методом.

Користувач вводить частини рівнянь системи, що містять в собі невідоме.

Програма обчислює корені системи рівнянь графічним методом, поступово зменшуючи інтервал побудови графіків

поблизу точки перетину функцій, та повертає відповідне значення округлене до двох знаків після коми,

а також на екрані виводиться графік перетину двох функцій.

Після чого користувач може ввести іншу систему рівнянь і програма почне свою роботу знову,

аж поки користувач не захоче вийти з програми.'''

requirements = '''

ВИМОГИ ДО ВВЕДЕННЯ

Для корректного вводу функції використовуйте наступні оператори:

"+" - оператор додавання

"-" - оператор віднімання

"\*" - оператор множення

"/" - оператор ділення.

"\*\*" - зведення в степінь, наприклад x\*\*3 - це х в кубі

"pi" - число пі. Наприклад, cos(x+pi/2)

log(x, e) - логарифм за основою e вираза в дужках. Наприклад, log(3\*x, 2) - логарифм з основою 2 числа 2x.

Другий параметр можна опустити, тоді це буде натуральний логарифм.

sin(x), cos(x), tan(x) - функції синус, косинус і тангенс від x відповідно.

arcsin(x), arccos(x), arctan(x) - функції арксинус, арккосинус і арктангенс від x відповідно.

Помітка: складні вирази потрібно брати в дужки'''

print(intoduction)

print('\_'\*30)

print(description)

print('\_'\*30)

print(requirements)

print('\_'\*30)

while True:

def check(f, x):

return (lambda x: f)(x)

try:

f = input("Введіть перше рівняння:\n")

f1 = input("Введіть друге рівняння:\n")

x = np.linspace(-10, 10, 201)

f\_1 = eval('lambda x: ' + f)

f\_2 = eval('lambda x: ' + f1)

y = f\_1(x)

y1 = f\_2(x)

k = 0

for i in range(len(y)):

#print(x[i], abs(y1[i] - y[i]))

if abs(y1[i] - y[i]) < 0.1:

print("Отримані корені:", "x =", round(x[i], 2), " f1(x) =", round(y[i], 2), " f2(x) =", round(y1[i], 2))

k += 1

if k==0:

print("Ця система рівнянь коренів не має!")

plt.plot(x, y, label=f)

plt.plot(x, y1, label=f1)

plt.legend()

plt.title("Система рівнянь")

plt.show()

except Exception:

print("Неправильний формат введених даних")

str = input("Якщо хочете продовжити роботу в програмі, введіть 'yes', в іншому випадку введіть будь-який символ: ")

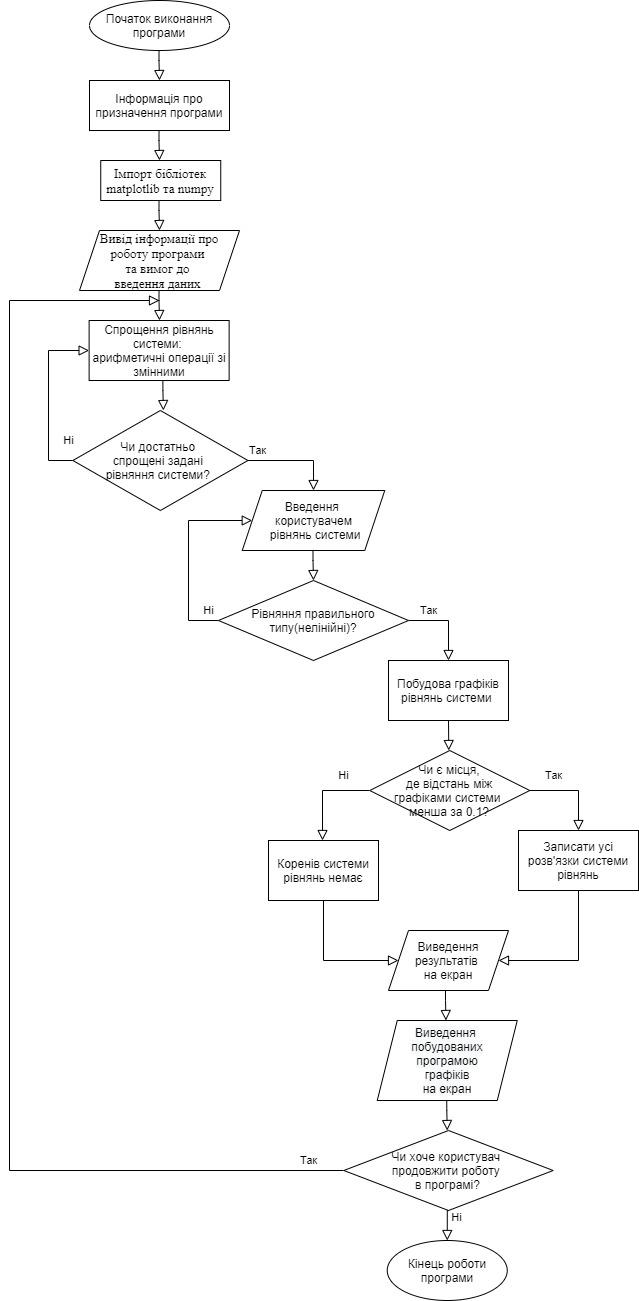
if str != 'yes':

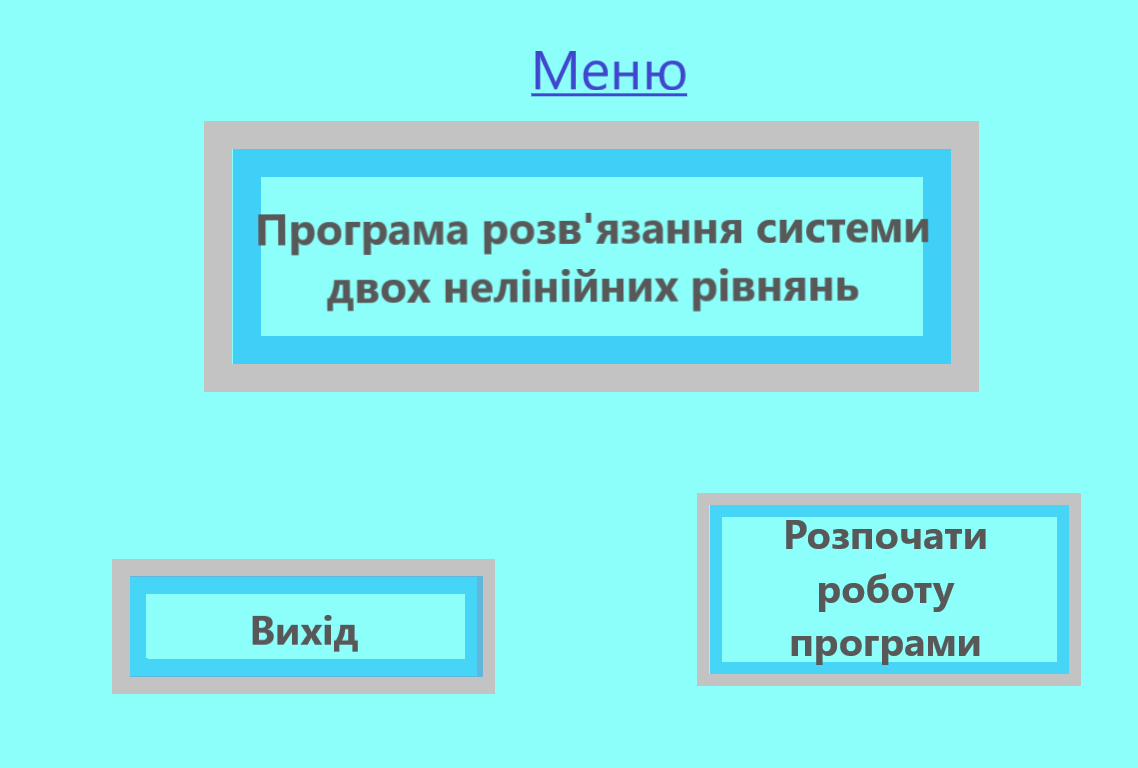
print('Програма завершує свою роботу!\nДо нових зустрічей!')

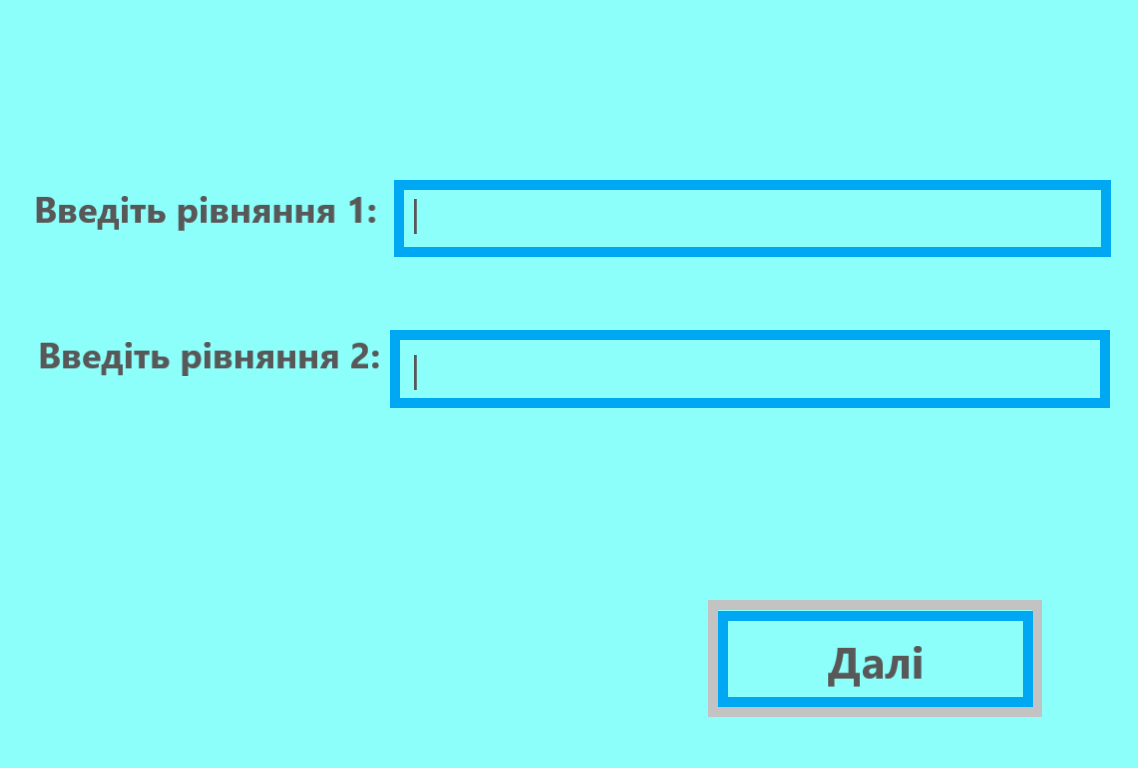
break

print()

# ДОДАТОК Б

 Блок-схема 1

****



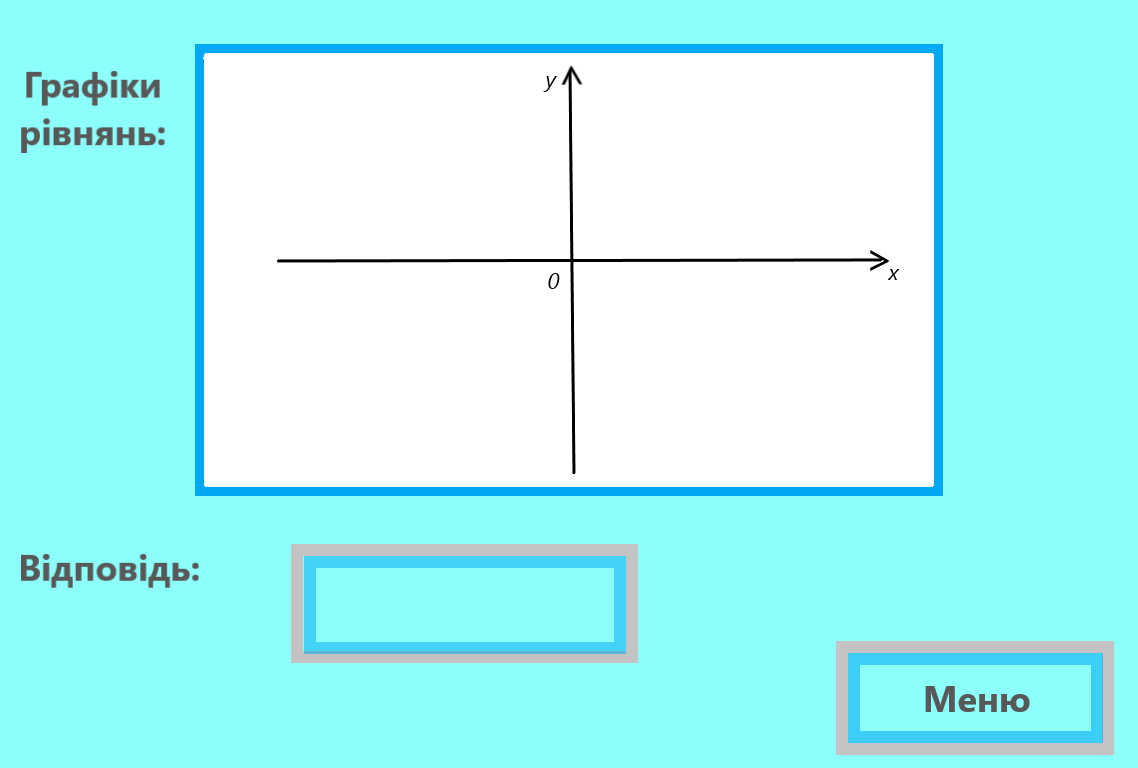
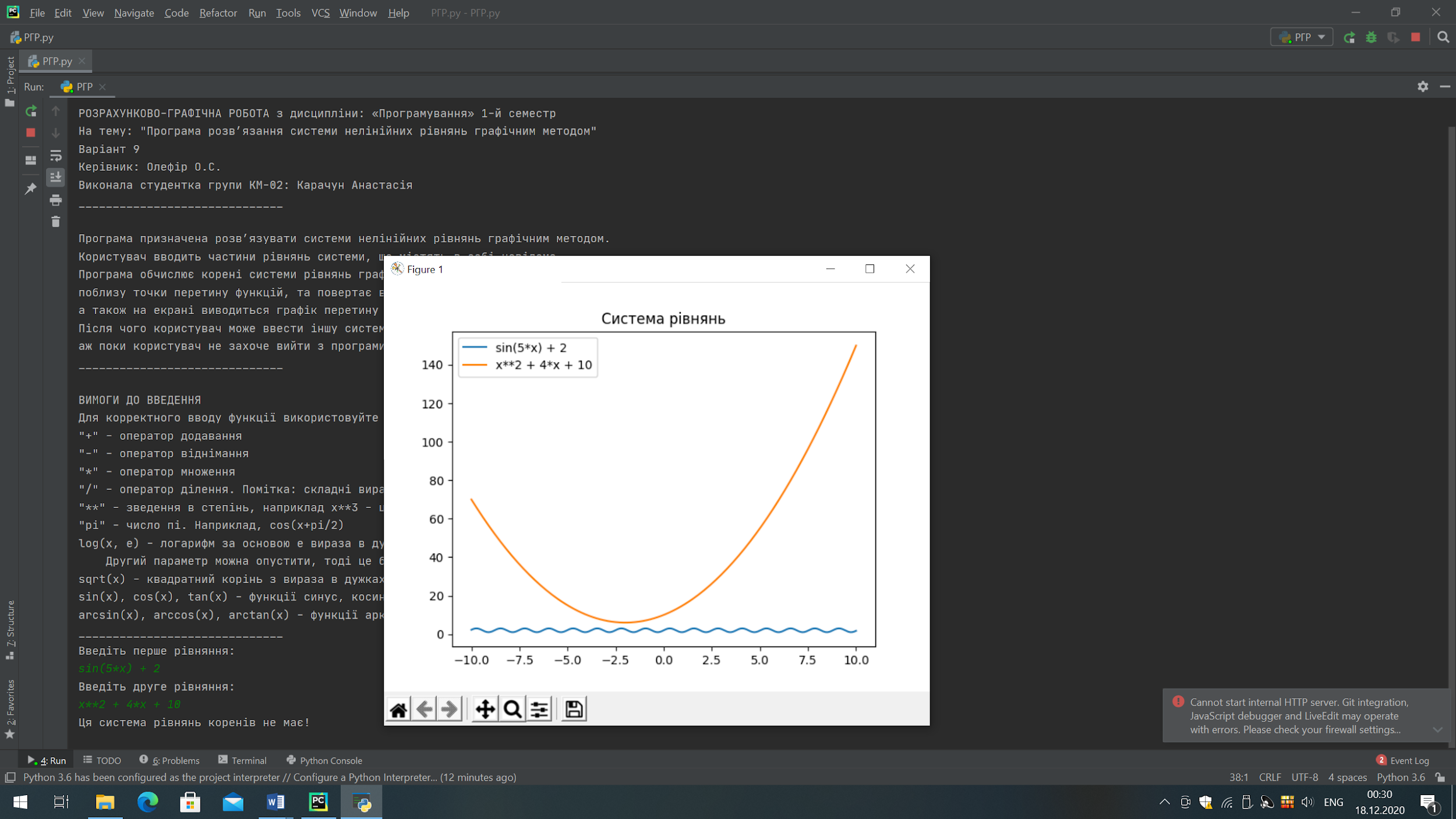
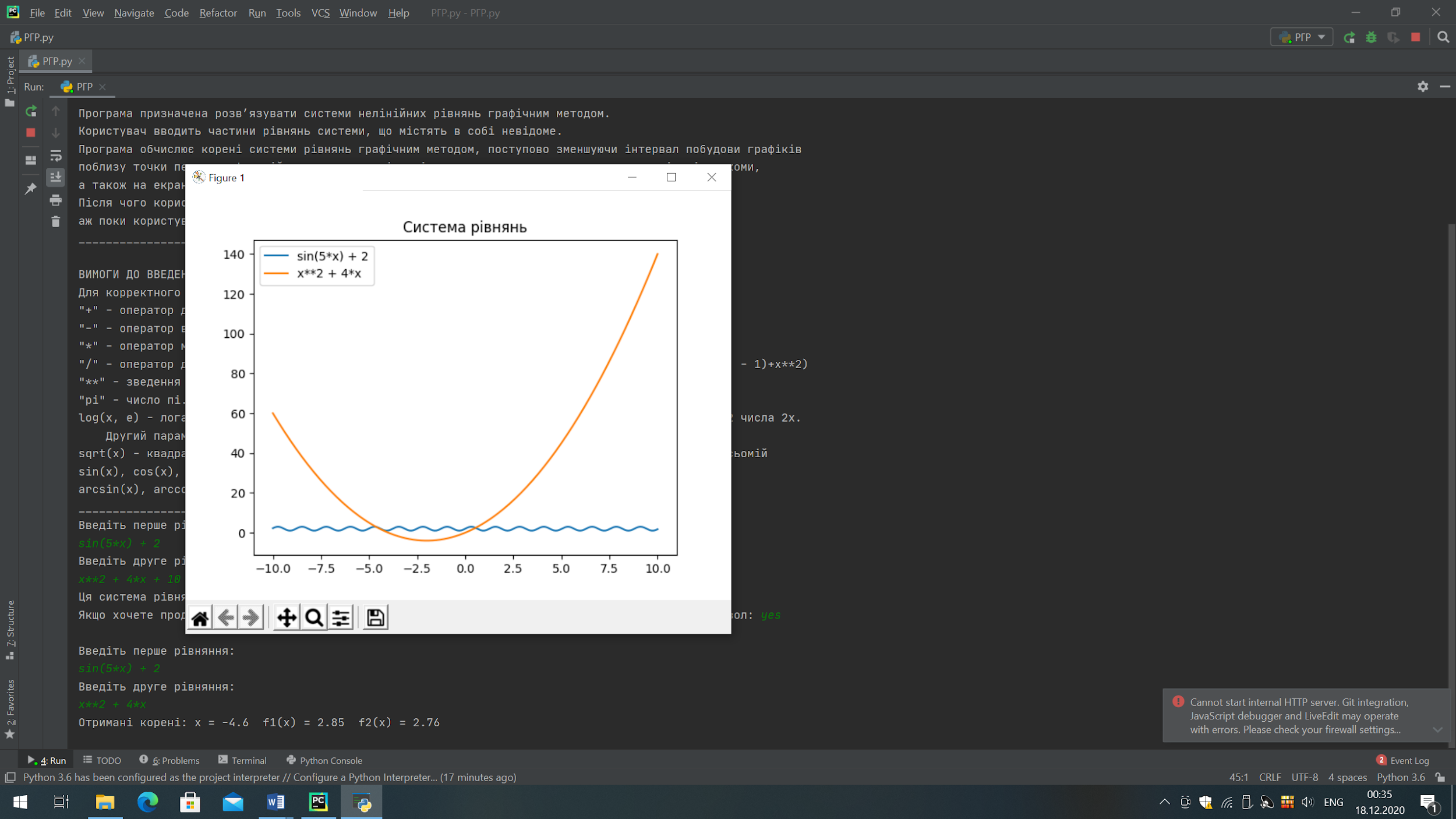
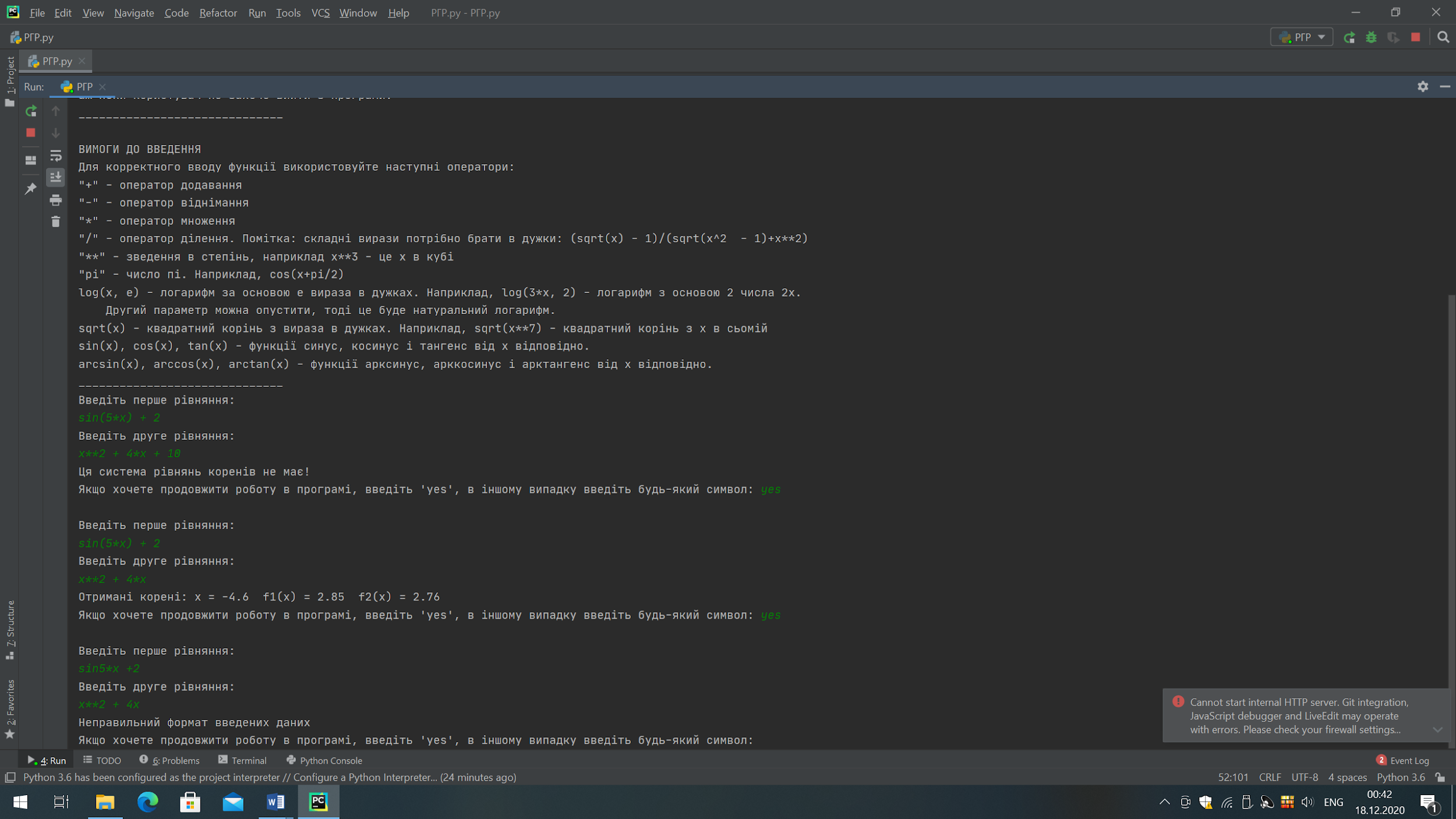
****

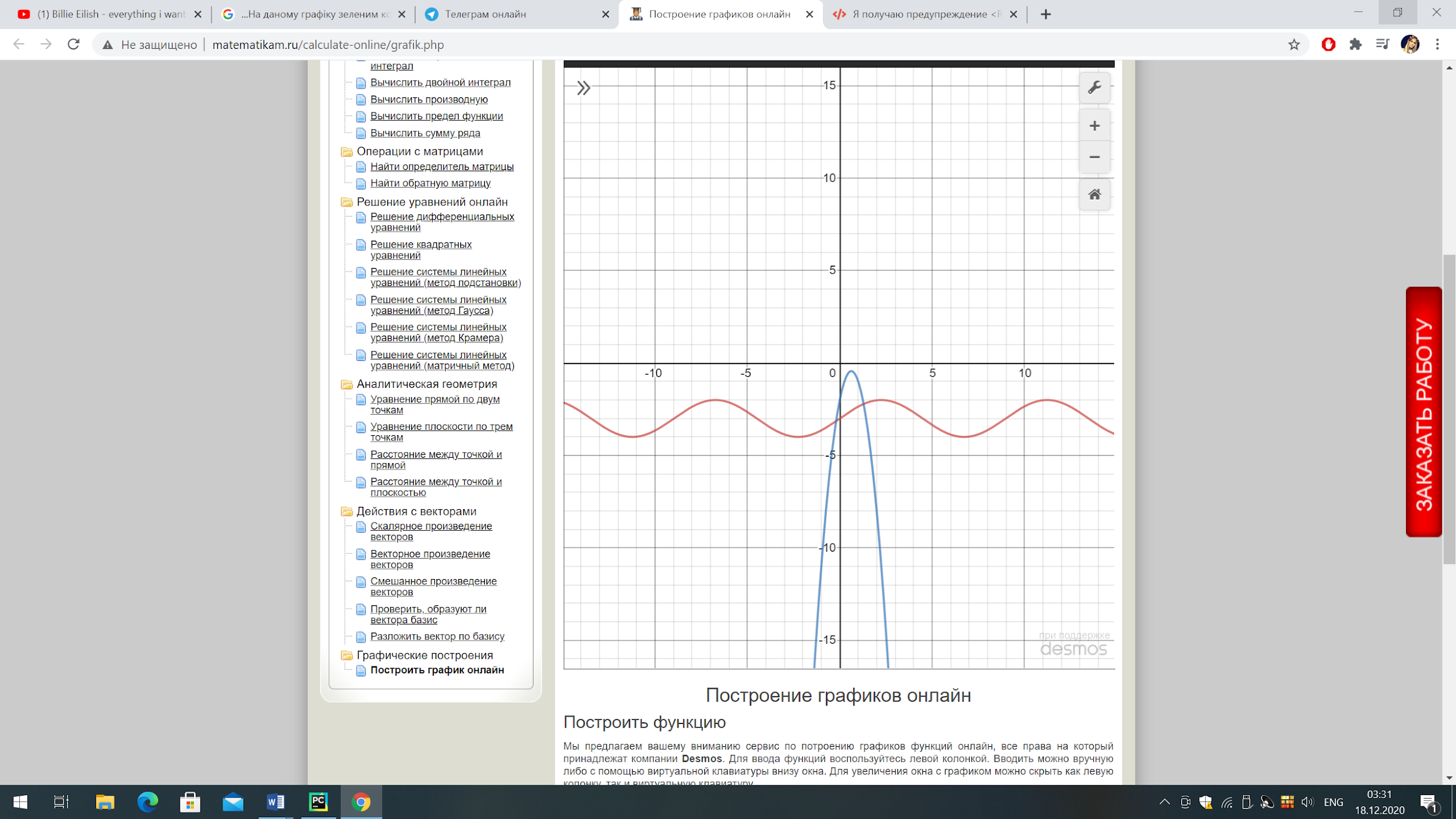
Рис. 2

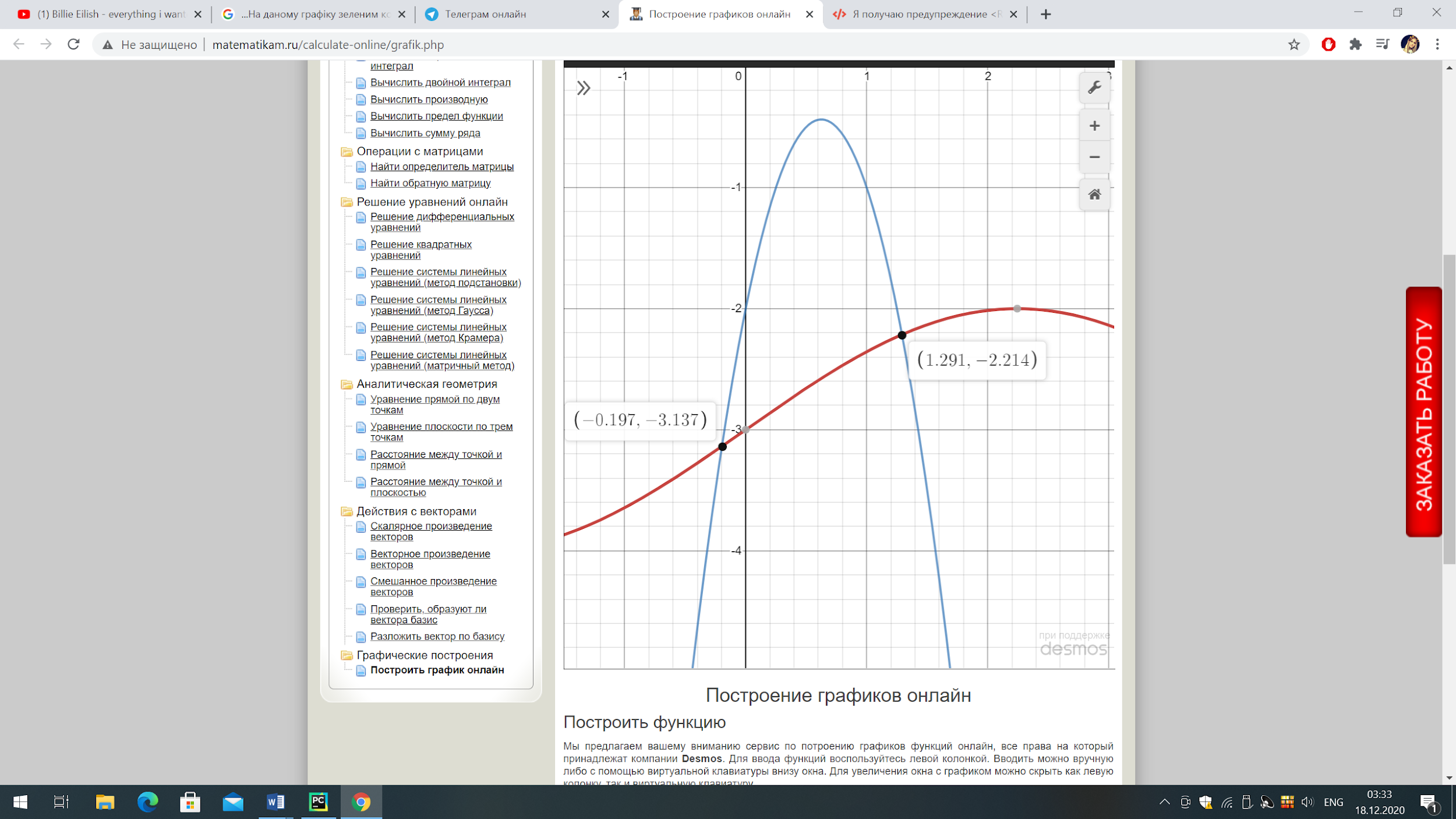
# ДОДАТОК В

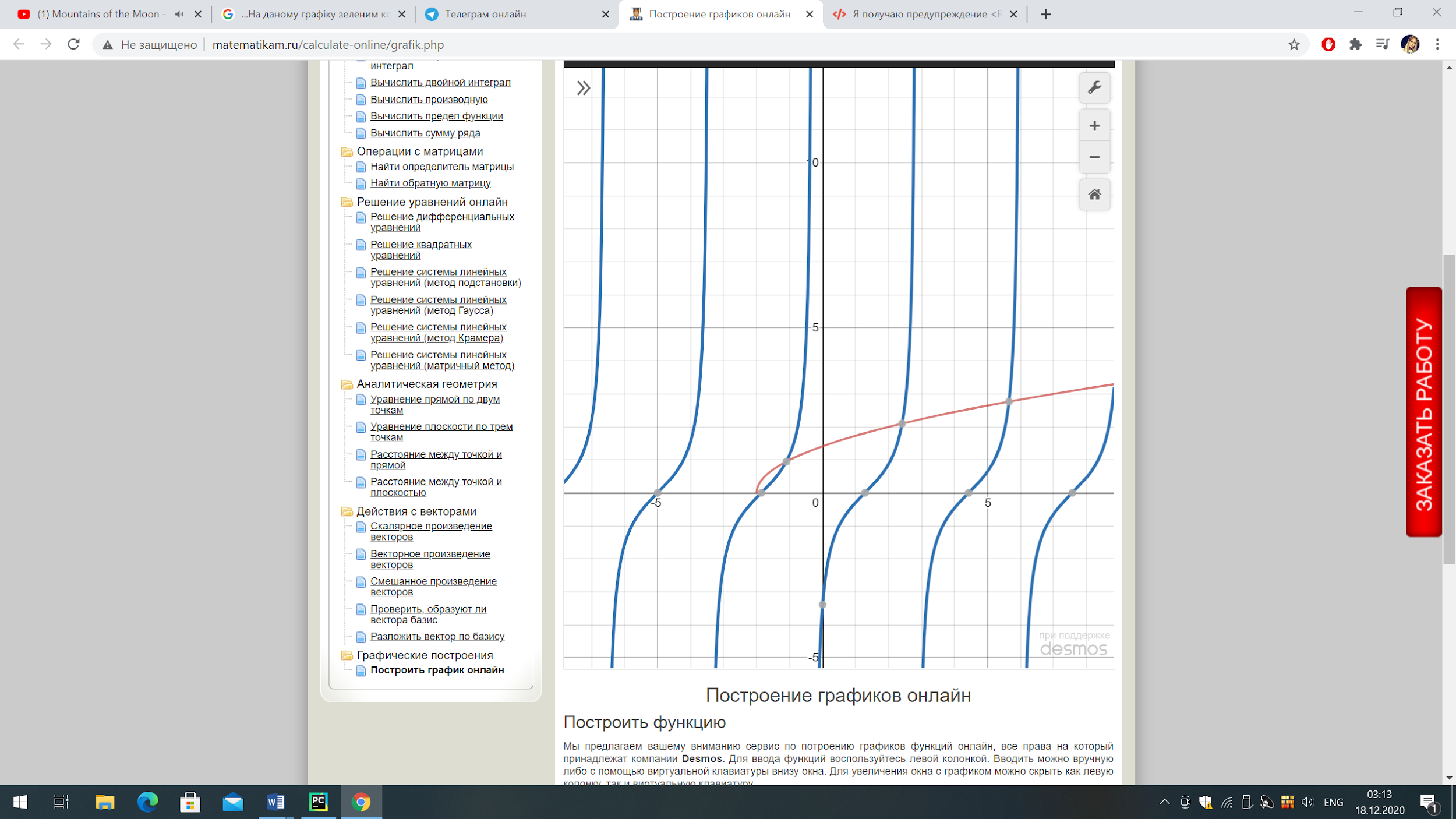
 Рис. 1 (а)

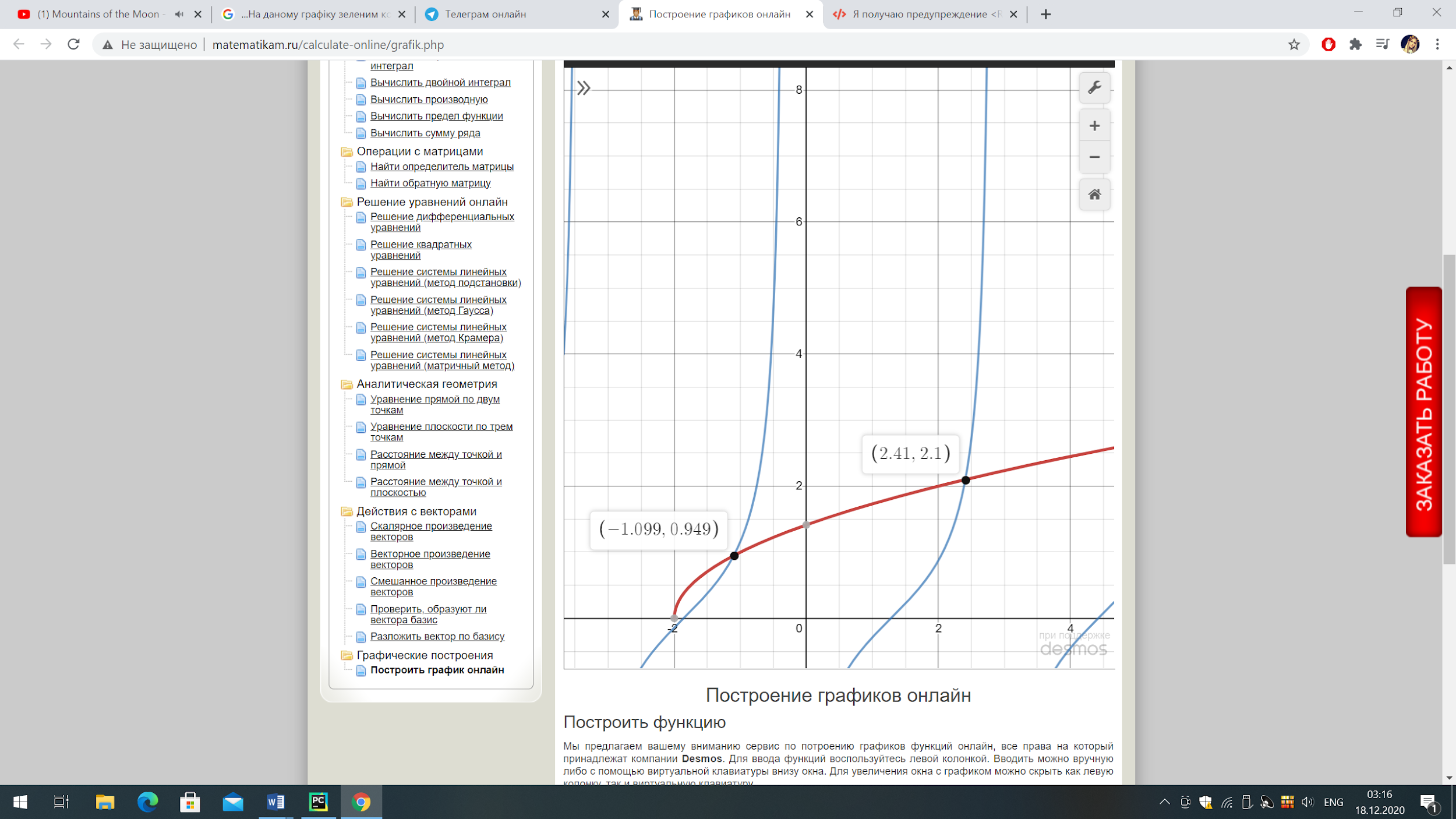
 Рис. 1 (б)

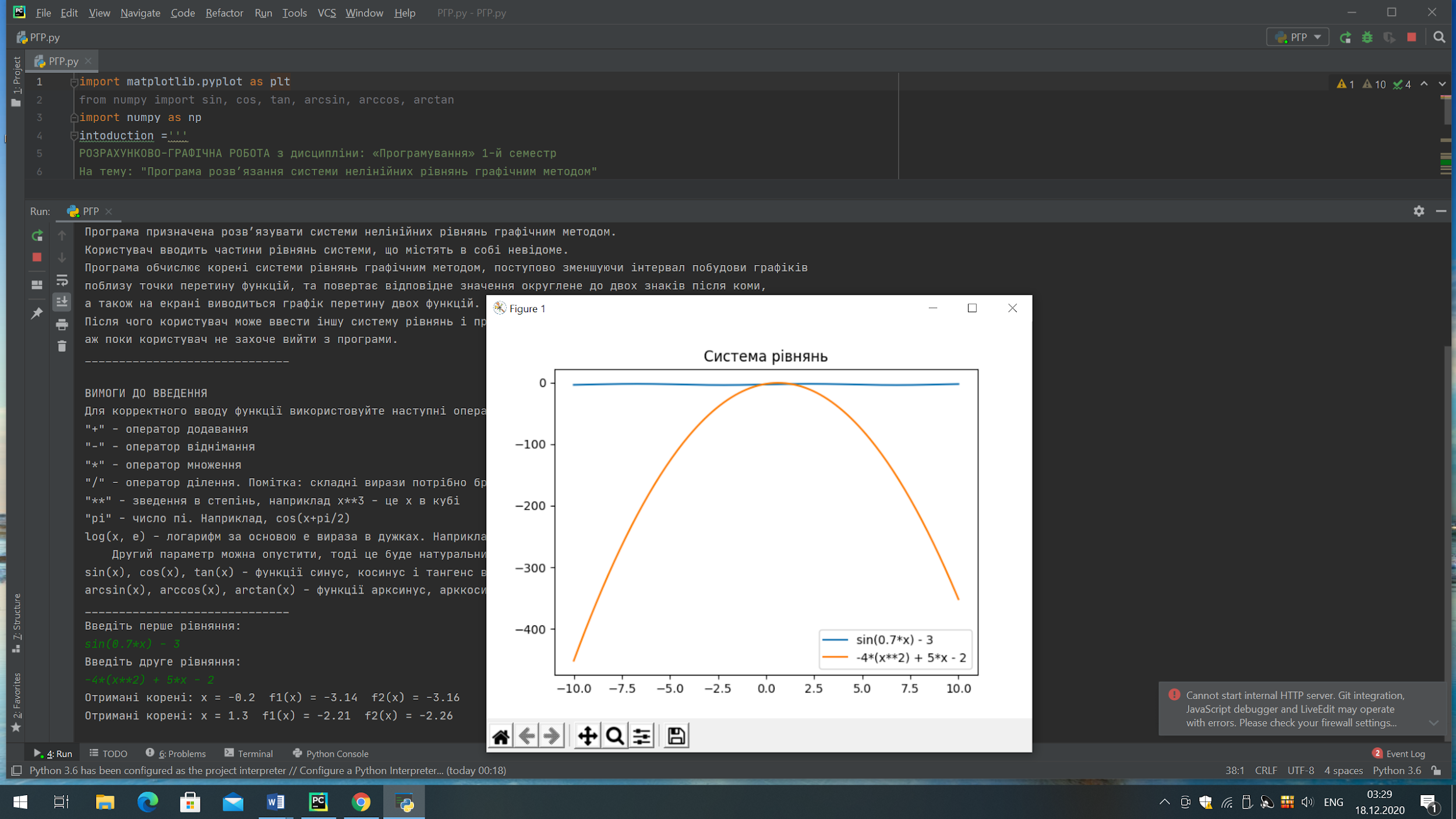
 Рис. 1 (в)

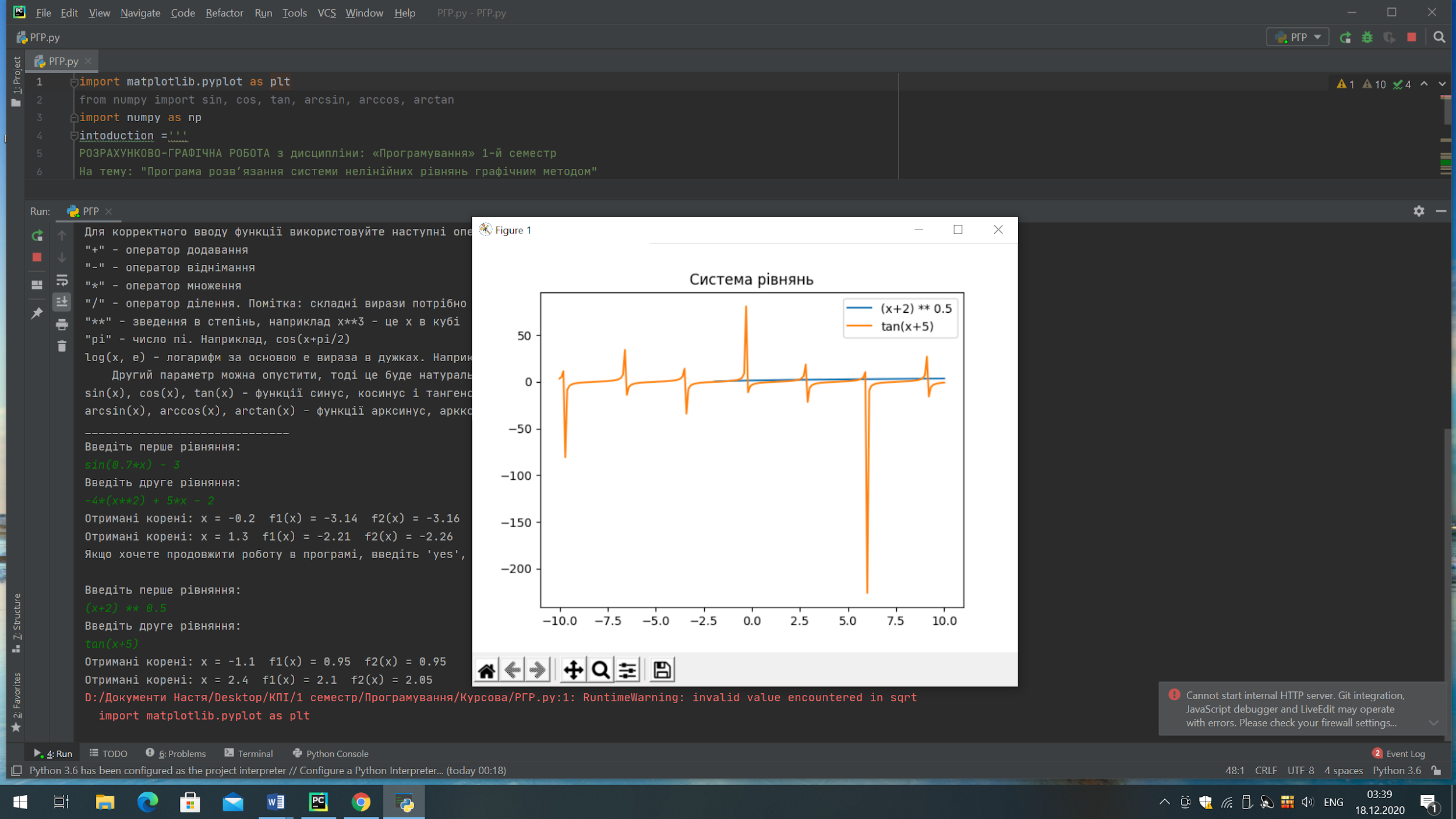
 Рис. 2 (а)

 Рис. 2 (б)

 Рис. 3 (a)

 Рис. 3 (б)

 Рис. 4 (а)

 Рис. 4 (б)