

Прізвище: Йовбак
 Ім'я: Андріанна
 Група: КНСП-11
 Варіант: 2
 Дата захисту: 12.11.2020



Кафедра: САПР
 Дисципліна: Методи та засоби комп'ютерного навчання
 Перевірив: Андрущак Н.А.

ЗВІТ

до лабораторної роботи №3 на тему «Побудова графіків в Python»

Мета роботи: розглянути можливості побудови графіків з використанням засобів представлення даних Python.

Відповіді на контрольні запитання:

- 1. Переваги пакету Matplotlib?** Переваги використання matplotlib в контексті аналізу і візуалізації даних полягає в наступному:
 - Побудовані дані є простими і інтуїтивно зрозумілим.
 - Висока продуктивність; професійний вихід.
 - Інтеграція з NumPy і SciPy (використовується для обробки сигналів і чисельного аналізу) не викликає труднощів.
 - Пакет налаштовується і конфігурується для потреби більшості людей.

Пакет досить великий і дозволяє, наприклад, вкладення ділянки в графічному інтерфейсі. В даний час пакет підтримує декілька графічних інтерфейсів, у тому числі WxPython і PyGTK.
- 2. Що таке інтерактивні графіки?** Існують 2 способи видимого відображення даних: інтерактивний графік або файли в друкованому вигляді. Інтерактивні графіки будують графік в окремому вікні за допомогою функції show().
- 3. Назвіть головні параметри збереження графіків у файл.** Функція savefig() дозволяє записувати зображення різних форматів в файл.

Параметр	Пояснення	Значення за замовчуванням	Приклад
dpi	Роздільна здатність в крапках на дюйм	Немає	savefig(fn, dpi=150)
facecolor*	Колір картини	'w' для білого кольору основи	savefig(fn, facecolor='b')
transparent	Коли картинка прозора	False	savefig(fn, transparent=True)
format	Формат файлу	'png'	savefig('image', format='pdf')

4. Які є доступні маркери при виводі графіків?

Характер	Маркер Символ
‘o’	Коло
‘^’	Трикутник
‘s’	Квадрат
‘+’	Плюс
‘x’	Хрест (множення)
‘D’	Ромб

5. **Як відбувається вивід підграфіків?** Для того аби відобразити декілька графіків на одній фігурі необхідно скористатися функцією subplot(). Підграфіки нумеруються зліва направо, зверху вниз, тому верхній лівий підграфік має значення 1, а нижній правий відповідає кількості графіків.
6. **Записати вираз $\frac{3\pi}{7}$.** Для коректного відтворення даного виразу потрібно здійснити такий запис: `r'$\frac{3\pi}{7}$'`
7. **Що таке гістограми, де вони використовуються?** Гістограми – графіки, які показують частоту або появу значень.
8. **Спосіб представлення кругових діаграм.** Кругові діаграми є простими у використанні, як гістограми та стовпцеві діаграми. Функція, яка реалізує кругові діаграми називається pie(x), де x є значення, яке буде представлятися.

Індивідуальне завдання:

Варіант 2.

Завдання №1

Згенерувати файл, в якому записані числові значення залежності інтенсивності сигналу I від частоти f в діапазоні від 10 ГГц до 70 ГГц з кроком 0.2 ГГц. Закон зміни інтенсивності: $I(f) = \sin(2\pi f t + \phi)$.

- вивести вміст файлу на екран;
- показати на графіку мінімальне та максимальне значення функції, перетину з осями координат, нулі функції;
- вивести на одному графіку два графіки залежності інтенсивності, які зсунуті по фазі на величину $\pi/2$;
- на одному графіку в різних вікнах зобразити: 1 – закон зміни інтенсивності сигналу; 2 – закон зміни інтенсивності з фазою $\pi/4$; 3 – закон зміни інтенсивності з фазою $\pi/8$;
- з використанням кругової діаграми, вивести кількість від'ємних та додатних значень закону зміни інтенсивності в діапазонах 10-30 ГГц, 31-50 ГГц, 51-70 ГГц.

Завдання №2

Для вибраного мікрорайону Львова (узгоджується індивідуально з викладачем) необхідно розробити програму, яка би:

- на карті в довільному порядку зобразити зупинки автобусів, маршрутних таксі та трамваїв. Кожен елемент необхідно представити різним кольором. Дані про розташування зупинок необхідно загрузити з файлу. У лівому верхньому куті вивести відповідність кольору до зупинки;

- б) на основі отриманих даних про зупинки, необхідно провести лінії відповідного кольору, які з'єднують зупинки;
- в) з використанням методів апроксимації, програмованими методами провести обвідну, яка проходить через вибрані точки;
- г) написати програму, яка би визначала відстань від однієї зупинки до іншої, в залежності від того, куди ми клацнули мишкою на карті з врахуванням існуючого маршруту.

Код програми:

```
# -----> LAB 3 <-----
from pylab import *
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import math

def task_1():
    #  $I(f) = \sin(2\pi f t + \phi)$ 
    def create_x(start, end, f_step):
        res = []
        f = start
        while f <= end:
            res.append(f)
            f += f_step
        return res

    def create_y(start, end, f_step, t, fi, write):
        if write:
            file = open('text.txt', 'w')
        arr_y = []
        f = start
        while f <= end:
            value = sin(2 * pi * f * t + fi)
            print(round(value, 1))
            if write:
                file.write(str(value)+'\n')
            arr_y.append(value)
            f += f_step
        return arr_y

    def count_negative_positive(start, end):
        pos = 0
        neg = 0
        result = []
        for i in create_y(start, end, 0.2, 1, 0, False):
            if i > 0:
                pos += 1
            if i < 0:
                neg += 1
        result.append(pos)
        result.append(neg)
        return result

    def convert_str_to_float(arr):
        result = []
        for i in arr:
            result.append(float(i))
        return result
```

```

# --- >> subtask 1 << ---
def sub_1():
    create_y(10, 70, 0.2, 1, 0, True)
    with open("text.txt") as values:
        values_arr = values.read()
    values_arr = values_arr[:-1]
    y = convert_str_to_float(values_arr.split('\n'))
    figure(1)
    plot(create_x(10, 70, 0.2), y, color="#ddaeff")
    show()

# --- >> subtask 2 << ---
def sub_2():
    figure(2)
    first_y = create_y(10, 70, 0.2, 1, 0, False)
    first_x = create_x(10, 70, 0.2)
    max_value = np.max(first_y)
    min_value = np.min(first_y)
    plot(first_x, first_y, color='#abf0d1')
    plot(first_x[first_y.index(max_value)], max_value, color='#58478d',
label='Max value', marker='o')
    plot(first_x[first_y.index(min_value)], min_value, color='#83052d',
label='Min value', marker='o')
    zero_x=[]
    zero_y=[]
    i = 10
    while i <= 70:
        zero_y.append(0)
        zero_x.append(i)
        i += 0.5
    plot(zero_x, zero_y, color='#589d7c', label='Нулі функції', marker='o',
linestyle='', ms=4)
    legend(loc=3)
    axhline(0, color='gray')
    axvline(0, color='gray')
    grid()
    show()

# --- >> subtask 3 << ---
def sub_3():
    first = create_y(10, 70, 0.2, 1, 0, False)
    second = create_y(10, 70, 0.2, 1, pi/2, False)
    figure(3)
    plot(create_x(10, 70, 0.2), first, label='sin(2πft)', color='#fe777e')
    plot(create_x(10, 70, 0.2), second, label='sin(2πft + π/2)',
color='#e0da48')
    legend(loc=1)
    show()

# --- >> subtask 4 << ---
def sub_4():
    first = create_y(10, 70, 0.2, 1, 0, False)
    second = create_y(10, 70, 0.2, 1, pi / 4, False)
    third = create_y(10, 70, 0.2, 1, pi / 8, False)
    x_a = create_x(10, 70, 0.2)
    figure(4)
    subplot(3, 1, 1)
    plot(x_a, first, label='sin(2πft)', color="#86b0f5")
    legend(loc=1)
    subplot(3, 1, 2)

```

```

plot(x_a, second, label='sin(2πft + π/4)', color="#ff93c9")
legend(loc=1)
subplot(3, 1, 3)
plot(x_a, third, label='sin(2πft + π/8)', color="#94ff7b")
legend(loc=1)
show()

```

```

# --- >> subtask 5 << ---

```

```

def sub_5():
    figure(5)
    # first pie chart start
    subplot(1, 3, 1)
    plt.title('From 10 ghz to 30 ghz', pad=30)
    res = count_negative_positive(10, 30)
    _, _, autotexts = pie(res, autopct='%1.1f%%',
        labels=[str(res[0]), str(res[1])],
        startangle=10, radius=1, colors=['#2db900', '#e80c6f'],
        explode=(0, 0.05))
    for autotext in autotexts:
        autotext.set_color('white')
    # first pie chart end
    # second pie chart start
    subplot(1, 3, 2)
    plt.title('From 31 ghz to 50 ghz', pad=-30, y=-0.01)
    res = count_negative_positive(31, 50)
    _, _, autotexts = pie(res, autopct='%1.1f%%',
        labels=[str(res[0]), str(res[1])],
        startangle=90, radius=1, colors=['#87cee7', '#c895c4'],
        explode=(0, 0.05))
    for autotext in autotexts:
        autotext.set_color('white')
    # second pie chart end
    # third pie chart start
    subplot(1, 3, 3)
    plt.title('From 51 ghz to 70 ghz', pad=30)
    res = count_negative_positive(51, 70)
    _, _, autotexts = pie(res, autopct='%1.1f%%',
        labels=[str(res[0]), str(res[1])],
        startangle=200, radius=1, colors=['#0e153b', '#af8706'],
        explode=(0, 0.05))
    for autotext in autotexts:
        autotext.set_color('white')
    # third pie chart end
    show()

```

```

sub_1()
sub_2()
sub_3()
sub_4()
sub_5()

```

```

def task_2():
    dots = []
    fig = plt.figure(figsize=(10, 8))
    global txt
    txt = None
    title('Select dots to calculate the distance', fontsize=10, pad=40)

    def draw_info_text():
        global txt

```

```

if txt is not None:
    txt.remove()
    fig.canvas.draw()
text_s = 'First dot: '
if len(dots) == 1:
    text_s += '-'
else:
    text_s += str(dots[len(dots) - 2])
text_s += '          Second dot: ' + str(dots[len(dots) - 1][0]) + \
'          Distance: ' + str(calc_distance())
txt = plt.text(0, -50, text_s, fontsize=10)
fig.canvas.draw()

def onclick(event):
    if event.xdata != None and event.ydata != None:
        print(event.xdata, event.ydata)

def calc_distance():
    if len(dots) == 0:
        return 0
    dot_1 = dots[len(dots) - 2][0]
    dot_2 = dots[len(dots) - 1][0]
    isTram = False
    isBus = False
    isTaxi = False
    global tram_stops
    if (dot_1[0] in tram_stops[0] and dot_2[0] in tram_stops[0]
        and dot_1[1] in tram_stops[1] and dot_2[1] in tram_stops[1]):
        isTram = True
    global bus_stops
    if (dot_1[0] in bus_stops[0] and dot_2[0] in bus_stops[0]
        and dot_1[1] in bus_stops[1] and dot_2[1] in bus_stops[1]):
        isBus = True
    global taxi_stops
    if (dot_1[0] in taxi_stops[0] and dot_2[0] in taxi_stops[0]
        and dot_1[1] in taxi_stops[1] and dot_2[1] in taxi_stops[1]):
        isTaxi = True
    if not isTaxi and not isBus and not isTram:
        return 'Dots from different routes'
    distance = 0
    if isTram:
        index = min(tram_stops[0].index(dot_1[0]),
tram_stops[0].index(dot_2[0]))
        while index < max(tram_stops[0].index(dot_1[0]),
tram_stops[0].index(dot_2[0])):
            index += 1
            distance += cal_dist_between_dots(tram_stops[0][index-1],
tram_stops[1][index-1],
                                                tram_stops[0][index],
tram_stops[1][index])
        if isBus:
            index = min(bus_stops[0].index(dot_1[0]), bus_stops[0].index(dot_2[0]))
            while index < max(bus_stops[0].index(dot_1[0]),
bus_stops[0].index(dot_2[0])):
                index += 1
                distance += cal_dist_between_dots(bus_stops[0][index-1],
bus_stops[1][index-1],
                                                    bus_stops[0][index],
bus_stops[1][index])
            if isTaxi:
                index = min(taxi_stops[0].index(dot_1[0]),

```

```

taxi_stops[0].index(dot_2[0]))
    while index < max(taxi_stops[0].index(dot_1[0]),
taxi_stops[0].index(dot_2[0])):
        index += 1
        distance += cal_dist_between_dots(taxi_stops[0][index-1],
taxi_stops[1][index-1],
                                         taxi_stops[0][index],
taxi_stops[1][index])
    return distance

def cal_dist_between_dots(x1, y1, x2, y2):
    dist = math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
    return dist

def onpick(event):
    thisline = event.artist
    xdata = thisline.get_xdata()
    ydata = thisline.get_ydata()
    ind = event.ind
    points = tuple(zip(xdata[ind], ydata[ind]))
    dots.append(points)
    draw_info_text()
    print('onpick points:', points)

img = mpimg.imread('map.png')
imgplot = plt.imshow(img)
plt.axis('off')
with open("stops.txt") as tstops:
    train_stops = tstops.read()
train_stops = train_stops.split('\n')
x = [float(row.split(',')[0]) for row in train_stops]
y = [float(row.split(',')[1]) for row in train_stops]
global tram_stops
tram_stops = [x[:5], y[:5]]
global taxi_stops
taxi_stops = [x[5:14], y[5:14]]
global bus_stops
bus_stops = [x[-5:], y[-5:]]
plot(x[:5], y[:5], marker='o', linestyle='-', picker=5, ms=8, color='#83052d',
label='tram stations')
plot(x[-5:], y[-5:], marker='o', linestyle='-', picker=5, ms=8,
color='#a75523', label='bus stations')
plot(x[5:14], y[5:14], marker='o', linestyle='-', picker=5, ms=8,
color='#58478d', label='taxi stations')
legend(loc=3)
cid = fig.canvas.mpl_connect('button_press_event', onclick) # значения
координат мышки
print('Mouse coordinats:', onclick)

fig.canvas.mpl_connect('pick_event', onpick)
plt.show()

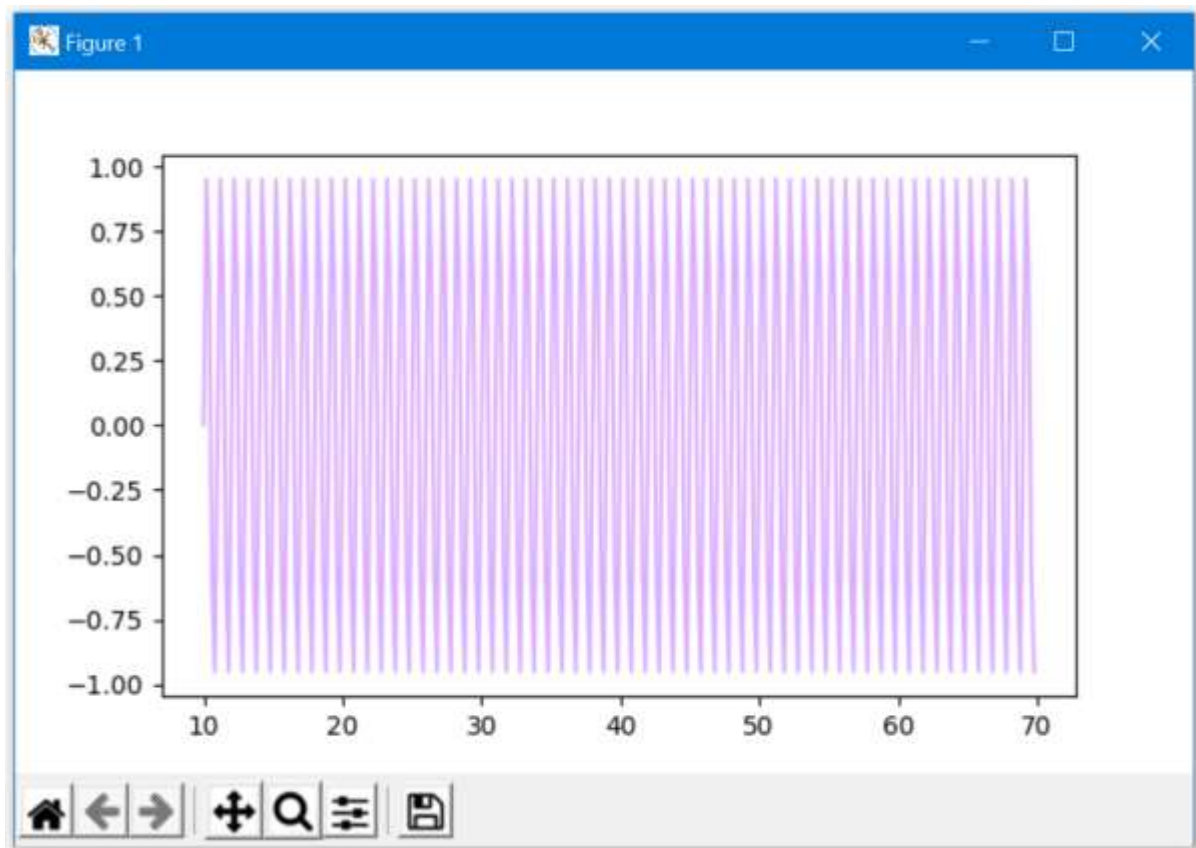
def main():
    task_1()
    task_2()

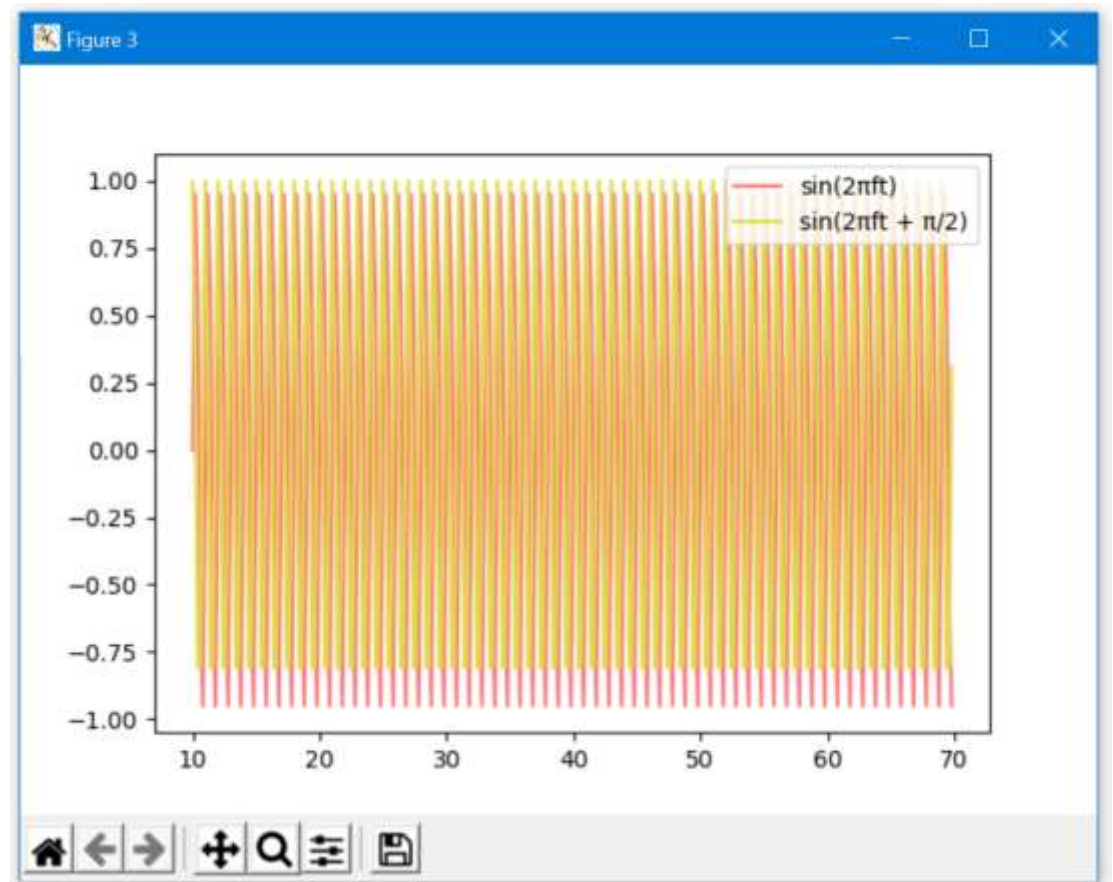
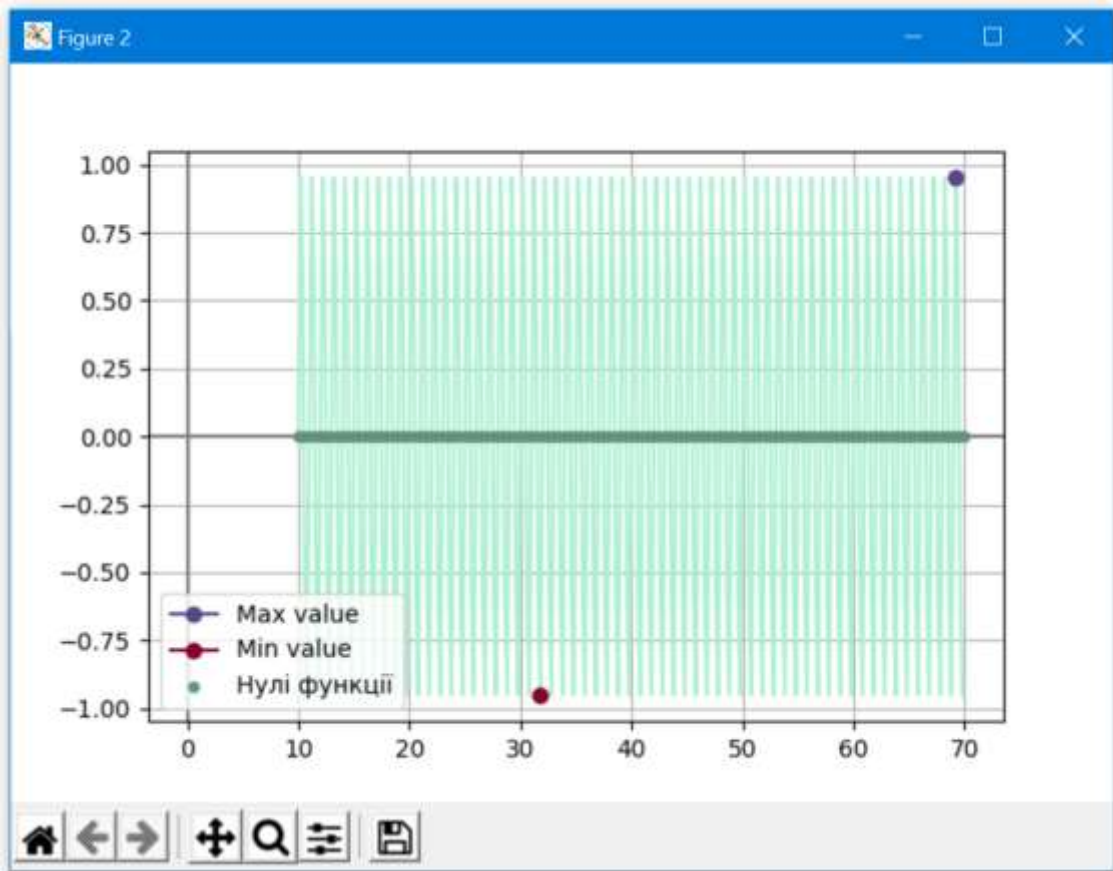
main()

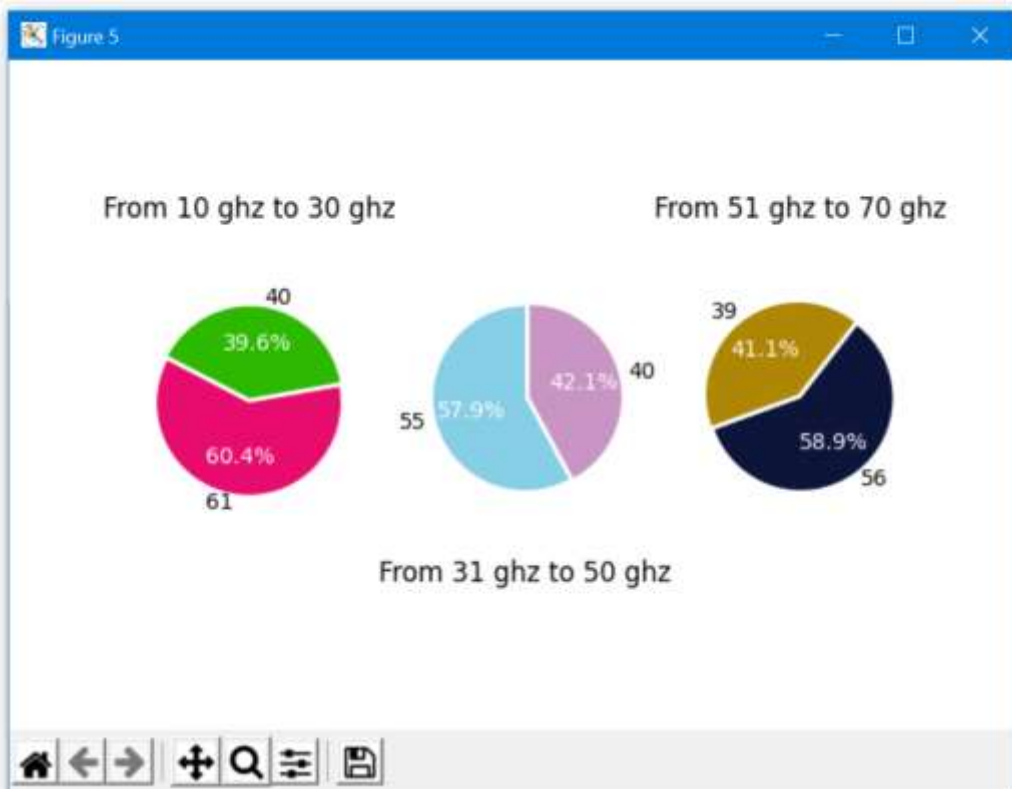
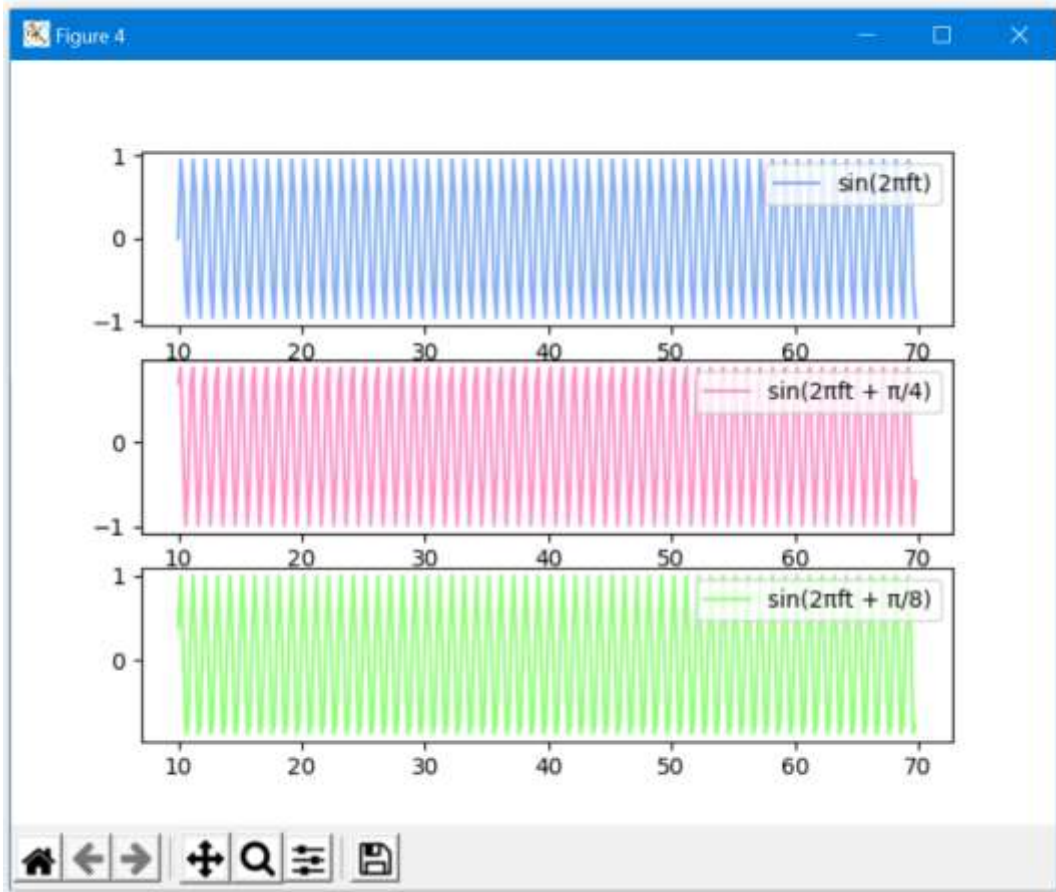
```

Результати виконання програми:

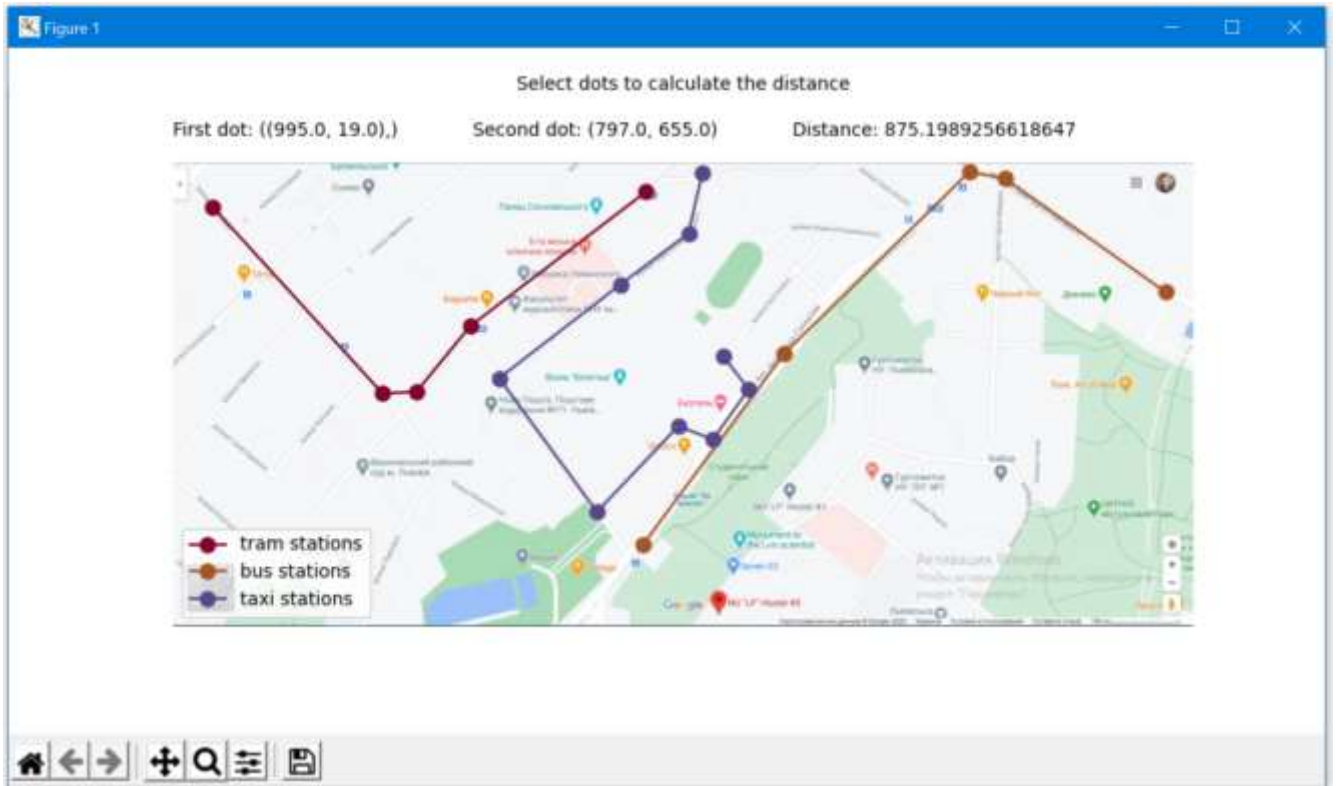
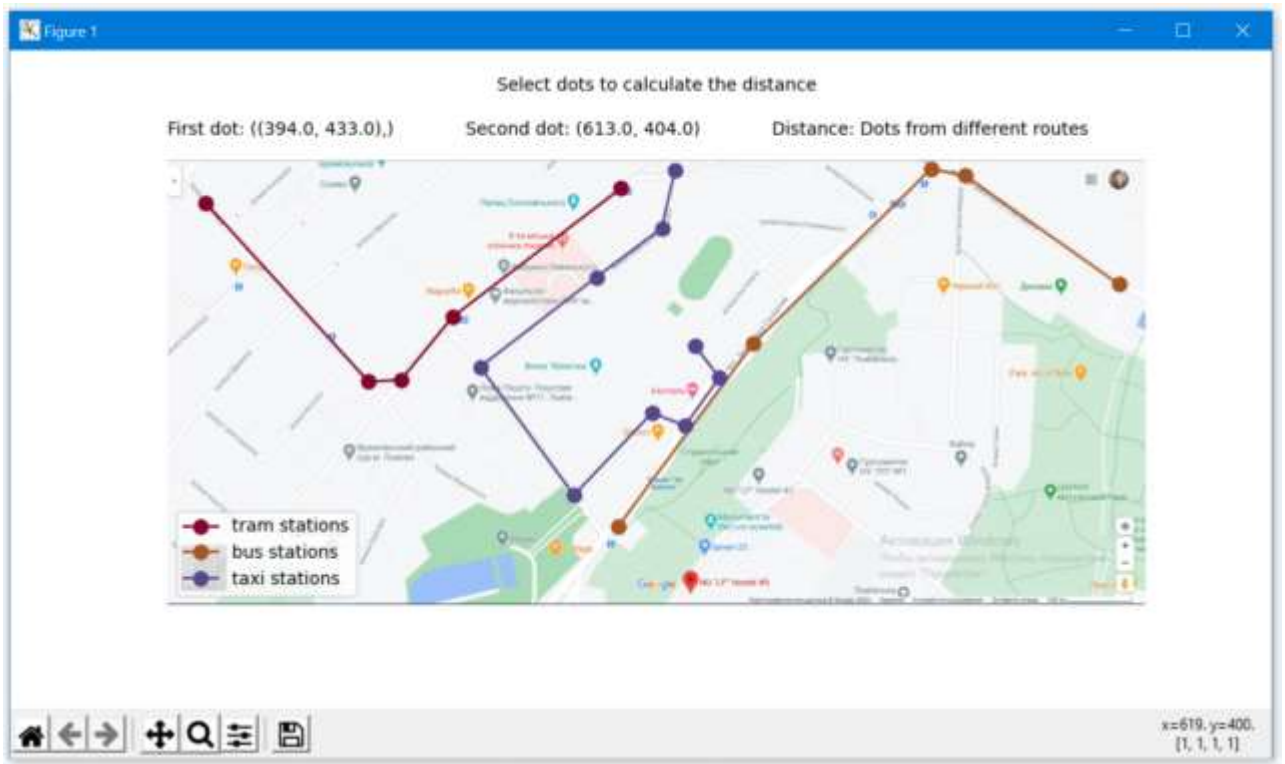
Завдання 1







Завдання 2



Аналіз результатів: В першому завданні створила функції для побудови необхідних масивів даних для відображення їх на інтерактивних графіках, також записала ці дані в файл, з якого їх потім зчитала. Також написала додаткові функції для обчислення значень з цих масивів, які є більше та менше нуля, функцію перетворення масиву стрічок в масив чисельних даних. В функціях `sub_1()`, `sub_2()`, `sub_3()`, `sub_4()`, `sub_5()` – написана логіка для виконання підпунктів а, б, в, г, д першого завдання відповідно. Виклик даних функцій відбувається в основній `task_1()`, яка їх згруповує.

В другому завданні вибрала район біля 5 гуртожитку НУЛП, в якому проживаю. Позначила основні автобусні, трамвайні маршрути та маршрути таксі, задавши відповідні точки в файлі `stops.txt`, з якого їх успішно зчитала при відображенні відповідних маркерів та доріг на карті. Також написала функцію для обчислення відстані між 2 зупинками одного маршруту. Якщо ж зупинки вибрані з різних маршрутів, виводиться відповідне повідомлення.

Висновок: на даній лабораторній роботі я розглянула можливості побудови графіків з використанням засобів представлення даних Python, а саме: зробила запис даних в файл, зчитала їх та побудувала інтерактивні графіки, кругові діаграми, навчилась їх стилізувати. Розглянула можливості та функції бібліотеки `matplotlib`.

Перелік використаних посилань:

1. Андрущак Н.А. Методи та засоби комп'ютерного навчання: лабораторний практикум для студентів другого (магістреського) рівня вищої освіти Інституту комп'ютерних наук та інформаційних технологій. – Львів: Видавництво Національного університету «Львівська політехніка», 2018. – 125 с.
2. Руководство по программированию на Python: <https://metanit.com/python/tutorial/>