Задание1

import folium

# Словарь с координатами стадионов

stadiums\_location = {

    "Лужники": (55.715551, 37.554191),

    "Спартак": (55.818015, 37.440262),

    "Динамо": (55.791540, 37.559809)

}

# Создаем карту Москвы

m = folium.Map(location=[55.7558, 37.6173], zoom\_start=11)

# Добавляем метки на карту

for stadium, coords in stadiums\_location.items():

    folium.Marker(location=coords, popup=stadium).add\_to(m)

# Сохраняем карту в файл

m.save("moscow\_stadiums\_map.html")

Задание 2

import folium

import math

# Функция для вычисления расстояния между двумя точками

def lonlat\_distance(a, b):

    degree\_to\_meters\_factor = 111 \* 1000  # 111 километров в метрах

    a\_lon, a\_lat = a

    b\_lon, b\_lat = b

    radians\_lattitude = math.radians((a\_lat + b\_lat) / 2.)

    lat\_lon\_factor = math.cos(radians\_lattitude)

    dx = abs(a\_lon - b\_lon) \* degree\_to\_meters\_factor \* lat\_lon\_factor

    dy = abs(a\_lat - b\_lat) \* degree\_to\_meters\_factor

    distance = math.sqrt(dx \* dx + dy \* dy)

    return distance

# Пример последовательности точек

points = [(37.6222, 55.7566), (37.6589, 55.7583), (37.6400, 55.7500)]

total\_distance = sum(lonlat\_distance(points[i], points[i + 1]) for i in range(len(points) - 1))

# Создаем карту и добавляем точки

m = folium.Map(location=[55.7558, 37.6173], zoom\_start=12)

for point in points:

    folium.Marker(location=(point[1], point[0])).add\_to(m)

# Средняя точка для метки

average\_point = (sum(p[1] for p in points) / len(points), sum(p[0] for p in points) / len(points))

folium.Marker(location=average\_point, popup="Средняя точка").add\_to(m)

# Сохраняем карту в файл

m.save("path\_map.html")

print(f"Общая длина пути: {total\_distance:.2f} метров")

Задание 3

import requests

def get\_satellite\_image(lat, lon):

    api\_key = 'YOUR\_API\_KEY'  # Вставьте ваш API ключ

    url = f"https://maps.googleapis.com/maps/api/staticmap?center={lat},{lon}&zoom=15&size=600x300&maptype=satellite&key={api\_key}"

    response = requests.get(url)

    if response.status\_code == 200:

        with open("satellite\_image.png", "wb") as f:

            f.write(response.content)

        print("Изображение сохранено как satellite\_image.png")

    else:

        print("Ошибка при получении изображения")

get\_satellite\_image(55.7558, 37.6173)

Задание 4

cities\_input = input("Введите города через запятую: ")

cities = [city.strip() for city in cities\_input.split(",")]

# Предположим, что у нас есть словарь с координатами городов

cities\_coordinates = {

    "Москва": (55.7558, 37.6173),

    "Сочи": (43.5853, 39.7202),

    "Калуга": (54.5127, 36.2636),

}

southernmost\_city = min(cities\_coordinates.keys(), key=lambda city: cities\_coordinates[city][0])

print(f"Самый южный город: {southernmost\_city}")

Задание 5

def find\_nearest\_pharmacy(address):

    api\_key = 'YOUR\_API\_KEY'  # Вставьте ваш API ключ

    geocode\_url = f"https://maps.googleapis.com/maps/api/geocode/json?address={address}&key={api\_key}"

    response = requests.get(geocode\_url)

    if response.status\_code == 200:

        location\_data = response.json()

        if location\_data['results']:

            location = location\_data['results'][0]['geometry']['location']

            lat, lon = location['lat'], location['lng']

            # Поиск аптек рядом с полученными координатами

            places\_url = f"https://maps.googleapis.com/maps/api/place/nearbysearch/json?location={lat},{lon}&radius=1500&type=pharmacy&key={api\_key}"

            places\_response = requests.get(places\_url)

            places\_data = places\_response.json()

            if places\_data['results']:

                nearest\_pharmacy = places\_data['results'][0]['name']

                print(f"Ближайшая аптека: {nearest\_pharmacy}")

            else:

                print("Аптеки не найдены.")

        else:

            print("Адрес не найден.")

    else:

        print("Ошибка при запросе.")

address\_input = input("Введите адрес: ")

find\_nearest\_pharmacy(address\_input)

Задание 6

import random

cities\_list = ["Москва", "Сочи", "Калуга"]

selected\_city = random.choice(cities\_list)

# Получаем координаты выбранного города для отображения на карте

city\_coordinates = {

    "Москва": (55.7558, 37.6173),

    "Сочи": (43.5853, 39.7202),

    "Калуга": (54.5127, 36.2636),

}

# Выбор случайного типа карты или спутника

map\_type = random.choice(["satellite", "roadmap"])

# Создаем карту с выбранным городом без названия

m = folium.Map(location=city\_coordinates[selected\_city], zoom\_start=12)

folium.Marker(location=city\_coordinates[selected\_city], popup="Угадай город!").add\_to(m)

m.save(f"{selected\_city}\_guess\_game\_map.html")

print(f"Показан участок карты города (название скрыто). Угадай город!")

Задание 7

def get\_district\_by\_address(address):

    api\_key = 'YOUR\_API\_KEY'  # Вставьте ваш API ключ

    geocode\_url = f"https://maps.googleapis.com/maps/api/geocode/json?address={address}&key={api\_key}"

    response = requests.get(geocode\_url)

    if response.status\_code == 200:

        location\_data = response.json()

        if location\_data['results']:

            district\_info = location\_data['results'][0]['address\_components']

            district\_name = next((component['long\_name'] for component in district\_info if 'administrative\_area\_level\_2' in component['types']), None)

            print(f"Район: {district\_name}")

        else:

            print("Адрес не найден.")

    else:

        print("Ошибка при запросе.")

address\_input = input("Введите адрес: ")

get\_district\_by\_address(address\_input)

задание 8

home\_address = input("Введите адрес дома: ")

university\_address = input("Введите адрес университета: ")

def get\_coordinates(address):

    api\_key = 'YOUR\_API\_KEY'

    geocode\_url = f"https://maps.googleapis.com/maps/api/geocode/json?address={address}&key={api\_key}"

    response = requests.get(geocode\_url)

    if response.status\_code == 200:

        location\_data = response.json()

        if location\_data['results']:

            return (location\_data['results'][0]['geometry']['location']['lng'],

                    location\_data['results'][0]['geometry']['location']['lat'])

        else:

            return None

coordinates\_home = get\_coordinates(home\_address)

coordinates\_university = get\_coordinates(university\_address)

if coordinates\_home and coordinates\_university:

    distance\_between\_points = lonlat\_distance(coordinates\_home, coordinates\_university)

    print(f"Расстояние от дома до университета: {distance\_between\_points:.2f} метров")

else:

    print("Не удалось получить координаты одного из адресов.")