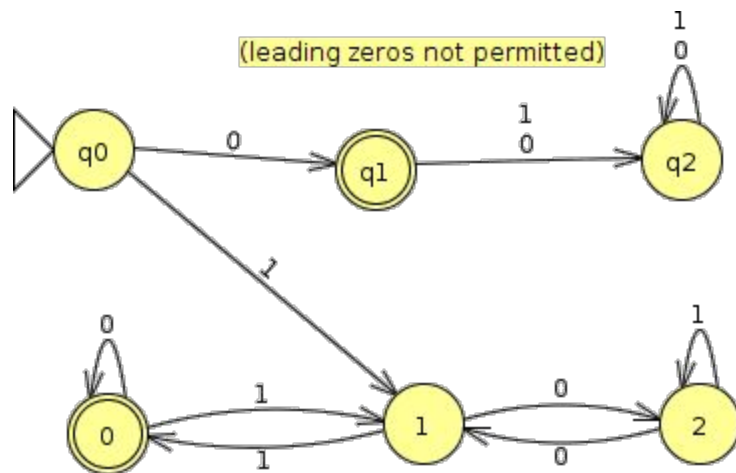


QUIZ

- 1) В файле, который я скинул, необходимо найти строчку, которая содержит подстроку "SPRING". Отправьте мне в лс эту строку и код, с помощью которого вы нашли эту строку
- 2) https://acmp.ru/index.asp?main=task&id_task=637

StatesGroup



Посмотрим код

```
18  class Form(StatesGroup):  
19      name = State()  
20      age = State()  
21      gender = State()  
22  
23  
24  
25  class Shop(StatesGroup):  
26      start = State()  
27      catalog = State()  
28      cart = State()  
29      address = State()
```

Подробнее о состояниях в aiogram

<https://mastergroosha.github.io/aiogram-3-guide/fsm/>

aiohttp

<https://docs.aiohttp.org/en/stable/>

```
2 import asyncio
3 import json
4
5 import aiohttp
6
7 from config import WEATHER_TOKEN
8
9 async def get_coord(city_name, token):
10     async with aiohttp.ClientSession() as session:
11         async with session.get(f"http://api.openweathermap.org/geo/1.0/direct?q="
12             f"{city_name},"
13             f"Russian Federation"
14             f"&appid={token}") as resp:
15             json_obj = json.loads(str(await resp.text()))
16             lat = str(json_obj[0]["lat"])
17             long = str(json_obj[0]["lon"])
18             return lat, long
```

```
20 async def get_weather(city, token):
21     lat, long = await get_coord(city, token)
22     print(lat, long)
23     async with aiohttp.ClientSession() as session:
24         async with session.get(f"https://api.openweathermap.org/data/2.5/weather?"
25                                f"lat={lat}"
26                                f"&lon={long}"
27                                f"&appid={token}&units=metric") as resp:
28             print(resp.status)
29             print(await resp.text())
30             json_obj = json.loads(str(await resp.text()))
31             temp = json_obj["main"]["temp"]
32             feels_like = json_obj["main"]["feels_like"]
33             humidity = json_obj["main"]["humidity"]
34             wind_speed = json_obj["wind"]["speed"]
35             print(temp, feels_like, humidity, wind_speed)
```


Подключимся к любой API и выгрузим какие-нибудь данные

```
40 async def get_currency(s):
41     async with aiohttp.ClientSession() as session:
42         async with session.get(f"https://api.exchangerate.host/latestbase=USD") as resp: # response
43             print(await resp.text())
44             json_obj = json.loads(str(await resp.text()))
45             return json_obj["rates"][s]
```

В этом нам помогут

https://catalog.api.2gis.ru/doc/2.0/transport/#/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82/get_2_0_transport_route_search

или

<https://yandex.ru/dev/rasp/>

DB, SQLAlchemy



I PREFER A REAL DATABASE



Excel

PERFECTION

```
1 from sqlalchemy import create_engine
2
3 from sqlalchemy import MetaData, Table, String, Integer, Column, Text, DateTime, Boolean
4 from datetime import datetime
5 from sqlalchemy import insert, select
6
7 metadata = MetaData()
8
9 blog = Table('blog', metadata,
10             Column('id', Integer()),
11             Column('name', String(50))
12             )
13
14 print(blog.columns)
15
16 engine = create_engine('sqlite:///sqlite3.db')
17
18 metadata.create_all(engine)
19
20 ins = blog.insert().values(
21     id="10",
22     name="Damir"
23 )
24 #
```

```
1 from sqlalchemy.ext.automap import automap_base
2 from sqlalchemy.orm import Session
3 from sqlalchemy import create_engine
4 from sqlalchemy.sql import select
5 from sqlalchemy import create_engine, MetaData, Table
6
7 engine = create_engine('sqlite:///sqlite3.db', echo=True)
8 Base = automap_base()
9 Base.prepare(engine, reflect=True)
10
11 # Users = Base.classes.Users
12 session = Session(engine)
13
14 metadata = MetaData()
15 blog = Table('blog', metadata,
16             autoload_with=engine)
17 cols = blog.c
18 |
19 ins = blog.insert().values(
20     id="11",
21     name="Insaf"
22 )
23
24 sel = select(blog).where()
25
26 conn = engine.connect()
27 conn.execute(ins)
28
29 query = blog.select()
30 result = conn.execute(query)
31
32 for row in result:
33     print(row)
```

Домашнее задание

- 1) Создать дб, в ней должна быть таблица с расписанием автобусов Москвы