**Lista Exercícios 4 - Lista Avançada**

**TODOS OS EXERCÍCIOS DEVEM SER FEITOS USANDO A TIPAGEM DAS VARIÁVEIS, INCLUSIVE NAS PASSAGENS DE PARÂMETRO (ENTRADA E RETORNO) -> Use o mypy para esse fim**

**USE O PYLINT PARA DEIXAR SEU PROGRAMA NO PADRÃO PEP8**

1. Faça um programa que leia o arquivo json abaixo e calcule o saldo diário da quantidade de moedas.
   1. O saldo inicial deve ser 0
   2. Se o campo type for igual “BUY” adiciona o qty ao saldo, se for “SELL” diminui qty do saldo.
   3. Cada linha do arquivo é uma transação
   4. Crie uma lista de Transações (transaction\_list: List[Transaction] = [] e jogue cada linha nessa lista (transation\_list.append).
   5. *A saída deve ser:*

*Saldo do dia 01-08-2022 => 100*

*Saldo do dia 02-08-2022 => 500*

*continua…*

* 1. *Exemplo da classe Transaction:*

@dataclass

class Transaction:

date: date

type: str

qty: float

price: float

cost: float

Arquivo de entrada: transactions.json

[{"date": "2022-08-01", "type": "BUY", "qty": 100.0, "price": 10.0, "cost": 1000.0}, {"date": "2022-08-02", "type": "BUY", "qty": 200.0, "price": 20.0, "cost": 4000.0}, {"date": "2022-08-02", "type": "BUY", "qty": 200.0, "price": 20.0, "cost": 4000.0}, {"date": "2022-08-03", "type": "BUY", "qty": 10.0, "price": 11.0, "cost": 110.0}, {"date": "2022-08-03", "type": "SELL", "qty": 15.0, "price": 11.0, "cost": 165.0}, {"date": "2022-08-03", "type": "SELL", "qty": 25.0, "price": 14.0, "cost": 350.0}, {"date": "2022-08-03", "type": "SELL", "qty": 300.0, "price": 14.0, "cost": 4200.0}, {"date": "2022-08-04", "type": "BUY", "qty": 25.0, "price": 11.0, "cost": 275.0}, {"date": "2022-08-04", "type": "BUY", "qty": 30.0, "price": 14.0, "cost": 420.0}, {"date": "2022-08-04", "type": "SELL", "qty": 5.0, "price": 11.0, "cost": 55.0}, {"date": "2022-08-04", "type": "SELL", "qty": 50.0, "price": 11.0, "cost": 550.0}, {"date": "2022-08-05", "type": "BUY", "qty": 10.0, "price": 12.0, "cost": 120.0}, {"date": "2022-08-05", "type": "SELL", "qty": 10.0, "price": 14.0, "cost": 140.0}, {"date": "2022-08-06", "type": "BUY", "qty": 100.0, "price": 12.0, "cost": 1200.0}, {"date": "2022-08-06", "type": "BUY", "qty": 80.0, "price": 10.0, "cost": 800.0}, {"date": "2022-08-06", "type": "SELL", "qty": 10.0, "price": 12.0, "cost": 120.0}, {"date": "2022-08-06", "type": "SELL", "qty": 90.0, "price": 12.0, "cost": 1080.0}, {"date": "2022-08-07", "type": "BUY", "qty": 20.0, "price": 12.0, "cost": 240.0}, {"date": "2022-08-07", "type": "SELL", "qty": 50.0, "price": 14.0, "cost": 700.0}, {"date": "2022-08-08", "type": "BUY", "qty": 20.0, "price": 12.0, "cost": 240.0}, {"date": "2022-08-08", "type": "BUY", "qty": 75.0, "price": 14.0, "cost": 1050.0}, {"date": "2022-08-08", "type": "SELL", "qty": 100.0, "price": 12.0, "cost": 1200.0}, {"date": "2022-08-09", "type": "SELL", "qty": 40.0, "price": 12.0, "cost": 480.0}]

1. Acrescente no exercício uma rotina para apresentar (no final) o total de compras (soma do qty de registro com type = ‘BUY’) e o total de vendas (soma do qty de registros com type = ‘SELL’)
2. Faça uma rotina que leia o arquivo do primeiro exercício da mesma forma (criando uma lista de transactions) mas faça uma lista ordenada (do maior para o menor) pelo campo cost. Imprima os campos cost, price, qty e date
3. Faça uma rotina que leia o arquivo do primeiro exercício da mesma forma e calcule a diferença de dias entre a menor data e a maior data.
4. Faça uma melhoria no programa 1 de forma a gerar um json com os saldos por dia. O arquivo json deve ser parecido com o exemplo abaixo. Crie uma classe chamada Balance com os campos date e saldo e gere o json a partir dela (use o comando json.dump com compreehension)

[

{"date": "2022-08-01", "saldo": 100.0},

{"date": "2022-08-02", "saldo": 4000.0},

...

]