link da aula do dia <https://meet.google.com/bjo-mumb-opm?pli=1&authuser=1>

desafio da aula de hoje foi desenvol uma aplicação usando método pilha e fila apreendidos

meu codigo:

class Hanoi:

def \_\_init\_\_(self, n):

self.torres = {

'A': list(range(n, 0, -1)),

'B': [],

'C': []

}

def push(self, destino, disco):

self.torres[destino].append(disco)

def pop(self, origem):

return self.torres[origem].pop() if self.torres[origem] else None

def mover(self, origem, destino):

if not self.torres[origem]:

print("Movimento inválido! Torre de origem vazia.")

return

disco = self.pop(origem)

if self.torres[destino] and disco > self.torres[destino][-1]:

print("Movimento inválido! Não pode colocar um disco maior sobre um menor.")

self.push(origem, disco)

else:

self.push(destino, disco)

def mostrar(self):

for t, pilha in self.torres.items():

print(f"{t}: {pilha}")

print()

n = int(input("Quantos discos? "))

jogo = Hanoi(n)

jogo.mostrar()

for \_ in iter(int, 1):

if jogo.torres['C'] == list(range(n, 0, -1)):

break

o, d = input("Mover de para (ex: A C): ").upper().split()

if o in 'ABC' and d in 'ABC':

jogo.mover(o, d)

jogo.mostrar()

else:

print("Entradas inválidas! Use A, B ou C.")

print("Parabéns! Você completou o jogo.")