class Item:

def \_\_init\_\_(self, name, price):

self.name = name

self.price = price

class Cart:

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

self.items = []

def add\_item(self, item):

self.items.append(item)

def remove\_item(self, item):

self.items.remove(item)

def calculate\_total(self):

return sum(item.price for item in self.items)

def apply\_discount(self, discount):

if self.calculate\_total() == 0:

raise ValueError("No se puede aplicar un descuento a un carrito de precio 0")

discounted\_total = self.calculate\_total() \* (1 - discount / 100)

if discounted\_total < 0:

raise ValueError("El descuento no puede resultar en un monto negativo")

return discounted\_total

def \_\_str\_\_(self):

items\_str = "\n".join([f"{item.name} - ${item.price}" for item in self.items])

return f"Carrito:\n{items\_str}\nTotal: ${self.calculate\_total()}"

# Ejemplo de uso

cart = Cart()

with open("items.txt", "r") as file:

while True:

line = file.readline()

if not line:

break

name, price = line.strip().split(",")

item = Item(name, float(price))

cart.add\_item(item)

print(cart)

try:

discounted\_total = cart.apply\_discount(10)

print(f"Total con descuento del 10%: ${discounted\_total}")

except ValueError as e:

print(f"Error: {str(e)}")