from collections import deque

# Array to store menu items

menu = []

# Queue to handle reservations

reservations = deque()

# Stack to manage customer feedback

feedback\_stack = []

# Array to store customer orders

orders = []

# Function to add a menu item

def add\_menu\_item():

    name = input("\nEnter menu item name: ")

    price = float(input("Enter menu item price: "))

    menu.append({"name": name, "price": price})

    print(f"Menu item '{name}' added successfully!")

# Function to display menu

def display\_menu():

    if not menu:

        print("\nThe menu is currently empty!")

        return

    print("\nRestaurant Menu:")

    for item in menu:

        print(f"{item['name']} - ${item['price']}")

# Function to sort the menu by price (Insertion Sort)

def sort\_menu\_by\_price():

    for i in range(1, len(menu)):

        key = menu[i]

        j = i - 1

        while j >= 0 and key["price"] < menu[j]["price"]:

            menu[j + 1] = menu[j]

            j -= 1

        menu[j + 1] = key

    print("\nMenu sorted by price!")

    display\_menu()

# Function to make a reservation

def make\_reservation():

    name = input("\nEnter customer name for reservation: ")

    reservations.append(name)

    print(f"Reservation added for {name}!")

# Function to process reservations

def process\_reservation():

    if reservations:

        print(f"\nProcessing reservation for {reservations.popleft()}")

    else:

        print("\nNo reservations to process!")

# Function to place an order

def place\_order():

    if not menu:

        print("\nMenu is empty! Add menu items before placing orders.")

        return

    display\_menu()

    name = input("\nEnter customer name: ")

    item = input("Enter menu item to order: ")

    for menu\_item in menu:

        if menu\_item["name"].lower() == item.lower():

            orders.append({"customer": name, "item": menu\_item})

            print(f"Order placed: {name} ordered {item}")

            return

    print("Menu item not found!")

# Function to display all orders

def display\_orders():

    if not orders:

        print("\nNo orders placed!")

        return

    print("\nOrders:")

    for order in orders:

        print(f"{order['customer']} ordered {order['item']['name']} (${order['item']['price']})")

# Function to sort orders by customer name (Bubble Sort)

def sort\_orders\_by\_customer():

    n = len(orders)

    for i in range(n):

        for j in range(0, n-i-1):

            if orders[j]["customer"].lower() > orders[j+1]["customer"].lower():

                orders[j], orders[j+1] = orders[j+1], orders[j]

    print("\nOrders sorted by customer name!")

    display\_orders()

# Function to add feedback

def add\_feedback():

    feedback = input("\nEnter customer feedback: ")

    feedback\_stack.append(feedback)

    print("Feedback added!")

# Function to view all feedback (LIFO order)

def view\_feedback():

    if not feedback\_stack:

        print("\nNo feedback available!")

        return

    print("\nCustomer Feedback (latest first):")

    for feedback in reversed(feedback\_stack):

        print(f"- {feedback}")

# Main menu

def main():

    while True:

        print("\n--- Restaurant Management System ---")

        print("1. Add Menu Item")

        print("2. Display Menu")

        print("3. Sort Menu by Price")

        print("4. Make Reservation")

        print("5. Process Reservation")

        print("6. Place Order")

        print("7. Display Orders")

        print("8. Sort Orders by Customer Name")

        print("9. Add Feedback")

        print("10. View Feedback")

        print("11. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":

            add\_menu\_item()

        elif choice == "2":

            display\_menu()

        elif choice == "3":

            sort\_menu\_by\_price()

        elif choice == "4":

            make\_reservation()

        elif choice == "5":

            process\_reservation()

        elif choice == "6":

            place\_order()

        elif choice == "7":

            display\_orders()

        elif choice == "8":

            sort\_orders\_by\_customer()

        elif choice == "9":

            add\_feedback()

        elif choice == "10":

            view\_feedback()

        elif choice == "11":

            print("Exiting...")

            break

        else:

            print("Invalid choice! Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()