from collections import deque

# List to store events

events = []

# Queue for event registrations

registrations = deque()

# Processed registrations counter

processed\_count = 0

# Graph for campus venues

venues = {

"Library": ["Auditorium", "Cafeteria"],

"Auditorium": ["Library", "Main Hall"],

"Cafeteria": ["Library", "Main Hall"],

"Main Hall": ["Auditorium", "Cafeteria"]

}

# Function to add an event

def add\_event():

print("\nAvailable Venues:")

for venue in venues:

print(venue)

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

date = input("Enter event date (YYYY-MM-DD): ")

venue = input("Enter event venue from the list above: ")

if venue not in venues:

print("Invalid venue! Event not added.")

else:

events.append({"name": name, "date": date, "venue": venue})

print("Event added successfully!")

# Function to remove an event

def remove\_event():

if not events:

print("No events available to remove!")

return

name = input("Enter the name of the event to remove: ")

for event in events:

if event["name"].lower() == name.lower():

events.remove(event)

print(f"Event '{name}' removed successfully!")

return

print("Event not found!")

# Function to search for an event by name

def search\_event():

name = input("Enter event name to search: ")

for event in events:

if event["name"].lower() == name.lower():

print(f"Event Found: {event['name']} on {event['date']} at {event['venue']}")

return

print("Event not found!")

# Function to sort events by date without modifying original order

def sort\_events():

if not events:

print("No events to sort!")

return

sorted\_events = sorted(events, key=lambda x: x["date"])

print("\nSorted Events by Date:")

for event in sorted\_events:

print(f"{event['name']} on {event['date']} at {event['venue']}")

# Function to display all events in the order they were added

def display\_events():

if not events:

print("No events to display!")

return

print("\nAll Events (Original Order):")

for event in events:

print(f"{event['name']} on {event['date']} at {event['venue']}")

# Function to find the shortest path between two venues

def find\_path():

print("\nAvailable Venues:")

for venue in venues:

print(venue)

start = input("\nEnter starting venue: ")

end = input("Enter destination venue: ")

if start not in venues or end not in venues:

print("Invalid venue names!")

return

# BFS for shortest path

queue = deque([[start]])

visited = set()

while queue:

path = queue.popleft()

node = path[-1]

if node == end:

print("Shortest path:", " -> ".join(path))

return

if node not in visited:

visited.add(node)

for neighbor in venues[node]:

new\_path = list(path)

new\_path.append(neighbor)

queue.append(new\_path)

print("No path found!")

# Function to register for an event with name and contact number

def register\_for\_event():

name = input("Enter your name to register: ")

contact = input("Enter your contact number: ")

registrations.append((name, contact))

print("Registration successful!")

# Function to process event registrations

def process\_registration():

global processed\_count

if registrations:

name, contact = registrations.popleft()

processed\_count += 1

print(f"Processing registration for {name} (Contact: {contact})")

else:

print("No registrations to process!")

# Function to view the registration status

def view\_registration\_status():

total\_registered = processed\_count

total\_pending = len(registrations)

print(f"\nRegistration Status:")

print(f" - Successfully Registered: {total\_registered}")

print(f" - Under Process (Pending): {total\_pending}")

# Function to count and display events on the same date

def count\_events\_by\_date():

if not events:

print("No events available!")

return

date = input("Enter the date (YYYY-MM-DD) to check events: ")

matching\_events = [event for event in events if event["date"] == date]

count = len(matching\_events)

if count == 0:

print(f"No events found on {date}.")

else:

print(f"\nThere are {count} event(s) scheduled on {date}:")

for event in matching\_events:

print(f" - {event['name']} at {event['venue']}")

# Main menu

def main():

while True:

print("\n--- Campus Event Management System ---")

print("1. Add Event")

print("2. Remove Event")

print("3. Search Event")

print("4. Sort Events by Date")

print("5. Display Events (Original Order)")

print("6. Find Path Between Venues")

print("7. Register for Event")

print("8. Process Registration")

print("9. View Registration Status")

print("10. Count Events on Same Date")

print("11. Exit")

choice = input("Enter your choice: ")

if choice == "1":

add\_event()

elif choice == "2":

remove\_event()

elif choice == "3":

search\_event()

elif choice == "4":

sort\_events()

elif choice == "5":

display\_events()

elif choice == "6":

find\_path()

elif choice == "7":

register\_for\_event()

elif choice == "8":

process\_registration()

elif choice == "9":

view\_registration\_status()

elif choice == "10":

count\_events\_by\_date()

elif choice == "11":

print("Exiting...")

break

else:

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

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

main()