using System;

using System.Collections.Generic;

using System.Linq;

using System.Text.RegularExpressions;

namespace \_04\_RoliTheCoder

{

public class RoliTheCoder

{

public static void Main()

{

var inputLine = Console.ReadLine();

var events = new Dictionary<string, Dictionary<string, List<string>>>();

var patternEvent = @"^#(.+)";

var eventRegex = new Regex(patternEvent);

var patternParticipant = @"^@([A-Za-z0-9'-]+$)";

var participantRegex = new Regex(patternParticipant);

while (inputLine != "Time for Code")

{

var line = inputLine

.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)

.ToArray();

var id = line[0];

var validParticipants = true;

var validEvent = eventRegex.IsMatch(line[1]);

if (validEvent)

{

for (int i = 2; i < line.Length; i++)

{

if (!participantRegex.IsMatch(line[i]))

{

validParticipants = false;

break;

}

}

}

if (validEvent && validParticipants)

{

var eventName = eventRegex.Match(line[1]).Groups[1].ToString();

var addParticipants = true;

if (!events.ContainsKey(id))

{

events[id] = new Dictionary<string, List<string>>();

events[id][eventName] = new List<string>();

}

else

{

if (!events[id].Keys.Contains(eventName))

{

addParticipants = false;

}

}

if (addParticipants == true)

{

for (int i = 2; i < line.Length; i++)

{

var nextParticipant = participantRegex.Match(line[i]).Groups[1].ToString();

if (!events[id][eventName].Contains(nextParticipant))

{

events[id][eventName].Add(nextParticipant);

}

}

}

}

events = events

.OrderByDescending(e => e.Value.First().Value.Count())

.ThenBy(e => e.Value.First().Key)

.ToDictionary(e => e.Key, e => e.Value

.ToDictionary(v => v.Key, v => v.Value.OrderBy(p => p).ToList()));

inputLine = Console.ReadLine();

}

foreach (var item in events.Values)

{

Console.WriteLine($"{item.First().Key} - {item.First().Value.Count()}");

foreach (var name in item.First().Value)

{

Console.WriteLine($"@{name}");

}

}

}

}

}