|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  namespace TeamworkProjects  {  class Program  {  static void Main(string[] args)  {  int teamsCount = int.Parse(Console.ReadLine());  var teamsList = new List<Teams>();  for (int i = 0; i < teamsCount; i++)  {  var input = Console  .ReadLine()  .Split("-")  .ToList();  string username = input[0];  string teamName = input[1];  Teams team = new Teams(username, teamName);  team.Users = new List<string>();  if (!teamsList.Select(x => x.TeamName).Contains(team.TeamName))  {  if (!teamsList.Select(x => x.Username).Contains(team.Username))  {  Console.WriteLine($"Team {teamName} has been created by {username}!");  teamsList.Add(team);  }  else  {  Console.WriteLine($"{username} cannot create another team!");  }  }  else  {  Console.WriteLine($"Team {teamName} was already created!");  }  }  string secondInput;  while ((secondInput = Console.ReadLine()) != "end of assignment")  {  var values = secondInput  .Split("->")  .ToList();  string username = values[0];  string teamName = values[1];  if (!teamsList.Select(x => x.TeamName).Contains(teamName))  {  Console.WriteLine($"Team {teamName} does not exist!");  }  else if (teamsList.Select(x => x.Users).Any(x => x.Contains(username)) ||  teamsList.Select(x => x.Username).Contains(username))  {  Console.WriteLine($"Member {username} cannot join team {teamName}!");  }  else  {  foreach (var team in teamsList)  {  if (team.TeamName == teamName)  {  team.Users.Add(username);  }  }  }  }  teamsList = teamsList  .OrderByDescending(x => x.Users.Count)  .ThenBy(x => x.TeamName)  .ToList();  foreach (var teams in teamsList)  {  if (teams.Users.Count >= 1)  {  Console.WriteLine(teams.TeamName);  Console.WriteLine("- " + teams.Username);  Console.WriteLine("-- " + string.Join(Environment.NewLine + "-- ", teams.Users));  }  }  Console.WriteLine("Teams to disband:");  foreach (var teams in teamsList)  {  if (teams.Users.Count == 0)  {  Console.WriteLine(teams.TeamName);  }  }  }  }  class Teams  {  public string Username { get; set; }  public string TeamName { get; set; }  public List<string> Users { get; set; }  public Teams(string username, string teamName)  {  this.Username = username;  this.TeamName = teamName;  }  }  } |