```
//
//  AppDelegate.swift
//  BansheeConsole
//
//  Created by Robert England on 9/13/15.
//  Copyright (c) 2015 Robert England. All rights reserved.
//

import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject:
        AnyObject]?) -> Bool {
        // Override point for customization after application launch.

        ////RE:: This should be all we need!
        let bansheeController = BansheeController()
        ////::RE

        return true
}   ...
```

```
    //
    //  Group.swift
    //  BansheeConsole
    //
    //  Created by Robert England on 9/13/15.
    //  Copyright (c) 2015 Robert England. All rights reserved.
    //
    //      MODEL: Manage info for one Kings Island visitor group
    //

    import Foundation

    //// info for one group
    struct Group: Printable {
        var name: String
        var town: String
        var size: Int

        // make this struct Printable: describe a Group
        var description: String {
            return "Group \(name) from \(town) has \(size) members\n"
        }

        // failable init: what if size isn't a number?
        init?(size: String, name: String, town: String) {
            if let sz = size.toInt() {
                if sz <= CoasterConstants.coasterCapacity {
                    self.size = sz
                    self.name = name
                    self.town = town
                }
                else {
                    println("(group is too big!)")
                    return nil   // init failed: group too big
                }
            }
            else {
                println("(nonnumeric group size)")
                return nil      // init failed: size isn't a number
            }
        }
    }
```

```swift
//
//  Coaster.swift
//  BansheeConsole
//
//  Created by Robert England on 9/13/15.
//  Copyright (c) 2015 Robert England. All rights reserved.
//
//      MODEL: Manage info for one roller coaster train
//

import Foundation

struct CoasterConstants {
    static let coasterCapacity = 32
}

//// info for one coaster
class Coaster: Printable {
    var ridingGroups = [Group]()
    var totalRiders = 0

    // make this class Printable: describe a Coaster
    var description: String {
        var s: String
        s = "\n\nCoaster with \(totalRiders) passengers:\n"
        for group in ridingGroups {
            s += "\t\(group)"
        }
        return s
    }

    // try to load a group on this coaster
    func load(group: Group) -> Bool {
        if group.size > CoasterConstants.coasterCapacity - totalRiders {
            return false
        }
        ridingGroups.append(group)
        totalRiders += group.size
        return true
    }
}
```

```swift
//
//  GroupCollection.swift
//  BansheeConsole
//
//  Created by Robert England on 9/13/15.
//  Copyright (c) 2015 Robert England. All rights reserved.
//
//      MODEL: Manage info for a collection of several groups
//

import Foundation

//// info for an array of groups
class GroupCollection {
    var groups = [Group]()

    // convert the raw group data into an array of Group structures
    init(dataArray groupDataArray: [String]) {
        var tempGroup: Group
        for var i = 0; i < groupDataArray.count; i += 3 {
            tempGroup = Group(size: groupDataArray[i],
                name: groupDataArray[i+1],
                town: groupDataArray[i+2])!
            groups.append(tempGroup)
        }
    }

    // add another group to the array of groups
    func append(newGroup: Group) {
```

```
            groups.append(newGroup)
        }

        // sort the groups into decreasing order by size
        func sort() {
            var swapped: Bool
            do {
                swapped  = false
                for var i = 0; i < groups.count-1; ++i {
                    if groups[i].size < groups[i+1].size {
                        let tempGroup = groups[i]
                        groups[i] = groups[i+1]
                        groups[i+1] = tempGroup
                        swapped = true
                    }
                }
            } while swapped
        }
    }
```

```
    //
    //  CoasterCollection.swift
    //  BansheeConsole
    //
    //  Created by Robert England on 9/13/15.
    //  Copyright (c) 2015 Robert England. All rights reserved.
    //
    //      MODEL: Manage infor for a collection of roller coaster trains
    //

    import Foundation

    //// info for an array of coasters
    class CoasterCollection: Printable {
        var coasters = [Coaster]()

        // make this Printable: describe a CoasterCollection
        var description: String {
            var s: String
            s = "\(coasters.count) Banshee trains are required:\n"
            for c in coasters {
                s += "\(c)"
            }
            return s
        }

        // load all groups from a GroupCollection wherever they'll fit
        func load(groupCollection: GroupCollection) {
            for g in groupCollection.groups {
                self.load(g)
            }
        }

        // load one group on a coaster, grabbing a new coaster if necessary
        func load(group: Group) {
            var loaded: Bool = false
            for c in coasters {
                if c.load(group) {
                    loaded = true
                    break
                }
            }
            if !loaded {
                var tempCoaster = Coaster()
                tempCoaster.load(group)
                coasters.append(tempCoaster)
            }
        }
    }
```

```swift
//
//  BansheeController.swift
//  BansheeConsole
//
//  Created by Robert England on 9/13/15.
//  Copyright (c) 2015 Robert England. All rights reserved.
//
//      CONTROLLER: This is a console application, so this class will play the role of
//                  Controller, and the Group, Coaster, etc. classes & structures will be the
//                  Model, and the humble console, bless its heart, will be our only View.
//


import Foundation

class BansheeController {

    // the MODEL for the Banshee app
    var groupCollection: GroupCollection        // The groups to ride the Banshee
    var coasterCollection: CoasterCollection    // The Banshee roller coaster trains

    init() {

        // hard code in (for now) the raw array of group data
        let groupDataArray = [
            "12","Girl Scouts","Newport, KY",               "9","Chocoholics","Berea, KY",
            "10","Diehard Reds Fans","Anderson, OH",        "3","Big Time Gamblers","Indianapolis, IN",
            "7","Dude Ranchettes","Houson, TX",             "30","Elvis Lives Club","Memphis, TN",
            "32","Transy Pioneers","Lexington, KY",         "2","C++ Hacker Guru Wizard
                Club","Lexington, KY",
            "21","Corvette Connoisseurs","Bowling Green, KY","20","Louisville Slugfest
                Club","Louisville, KY",
            "20","Polar Divers","Reykjavik, Iceland",       "4","Doris Day Fan Club","Cincinnati, OH",
            "14","Dog Walkers","Bracktown, KY",             "30","Shameless Squaredancers","Cincinnati,
                OH",
            "12","Tweaking Twangers","Nashville, TN",       "32","Big Blue Nation","Lexington, KY",
            "1","Lonely Hearts Club","Waddy, KY"]

        // create the list of group structures
        groupCollection = GroupCollection(dataArray: groupDataArray)

        // sort the groups, make a coaster collection, put the groups in coasters,
        //    and print the loading schedule to the console
        // Ba-da boom, Ba-da bing.
        groupCollection.sort()
        coasterCollection = CoasterCollection()
        coasterCollection.load(groupCollection)
        println("\(coasterCollection)")
    }

}
```