函数式语言程序设计

Homework 2

**作业要求**

1. 请为每个题目单独创建一个文件夹，文件夹名字与题目名字保持一致。每个题目的文件夹内包含一个名为“**Solution.hs**”的文件（S是大写），文件内包含一个名为“**solution**”的函数（小写s）。函数solution的**类型**必须严格与题目要求中一致，否则影响得分。文件中可以自由定义其他辅助函数。
2. 将所有题目的文件夹压缩为一个 “**zip**” 文件，并使用“**YourName\_StuNum.zip**”的形式命名该文件。其中YourName是名字的拼音，StuNum是学号，中间用下划线分割。请注意，zip文件需要**直接**包含所有题目的目录，而不应包含多余的目录层次（参考例子）。

例如，假定完成了 “p1” 和 “p2” 两个题目，则提交时的目录结构为：

YourName\_StuNum.**zip**

|- p1 (the problem name, given in the problem description.)

\- Solution.hs

\- p2

\- Solution.hs

每个 “.hs” 文件中都有solution 函数

module Solution where

-- Auxiliary functions

-- Main function

**solution** = …

**Problem 1: list-equal**

给定两个类型相同的列表，判断两个列表是否相等。禁止使用内置的相等比较，即 [1, 2] == [1, 2] 这种形式。实现中列表类型为 [Integer]

示例：

ghci> :t solution

solution :: [Integer] -> [Integer] -> Bool

ghci> solution [1] [2]

False

ghci> solution [1, 2, 3] [1, 2, 3]

True

**Problem 2: couples**

现在有两支队伍进行比赛，当然一对穿的是红色队服，一队穿的是蓝色队服啦！每个队的队员都有一个数字号码用来标识身份，一个队伍内的队员号码是彼此不同的。比赛开始后发生了一件奇怪的事情，红队中号码和蓝队中号码相同的队员两两消失在了天际，他们去了哪里我们不得而知，后世的人们只留下了“自古红蓝出CP（couple）”的传说！

给定两支队伍的队员号码列表（列表长度有限，且两支队伍人数相同，每支队伍内队员号码彼此不同，且列表中号码按照升序给出），求队员配对消失后场上剩余队员的数目。

示例：

ghci> :t solution

solution :: [Integer] -> [Integer] -> Integer

ghci> solution [1, 2, 3] [1, 3, 4]

2

ghci> solution [1, 2, 3] [1, 2, 3]

0