# Problem 4 – RainAir

Before naming it RyanAir … Tony Ryan named it RainAir, because the day he named it, it was really rainy, and he liked rain. Anyways, you have been hired by Tony, to create a software which manipulates data about flights and customers. The future of RyanAir is in your hands.

You will receive input lines in one of the following formats:

* {customerName} {customerFlight1} {customerFlight2} {customerFlight3} ...
* {customerName} = {customer2Name}

The customerName is a string. The customerFlights are integers.

If you receive a customerName and customerFlights, you should **add the customer** and **the flights** to the customer.   
If the customer **already exists**, just **add** the **new flights** to him.

If you receive a customerName and customer2Name, you should **make** the **1st customer**’s flights **equal** to the **2nd customer**’s flights.

The input ends when you receive the command “I believe I can fly!”. When that happens, you must **print all customers**, **ordered** by **count** of **flights** in **descending order**, and then by **alphabetical order**.

The **flights** must be ordered in **ascending order**.

## Input

* The input consists of several input lines in the format specified above.
* The input ends when you receive the command “I believe I can fly!”.

## Output

* As output you must print all the customers ordered in the way specified above.
* The format is: #{customerName} ::: {flight1}, {flight2}, {flight3}...

## Constraints

* There will be **no invalid input lines**.
* The customerName is a string which may contain **any ASCII characters** **except** ‘ ’ (space) and ‘=’.
* The customerFlight is an integer in **range [0, 10000]**.
* There will be **no non-existent** customerNames in the commands that require customerNames.
* If **all data ordering fails**, you should **order** the data by **order** **of** **input**.
* Allowed working **time** / **memory**: **100ms** / **16MB**.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Donald 1549 4592 3945 111  Prakash 111 45  Gibbs 492 502  Isacc 204 544  I believe I can fly! | #Donald ::: 111, 1549, 3945, 4592  #Gibbs ::: 492, 502  #Isacc ::: 204, 544  #Prakash ::: 45, 111 |
| Prakash 111 134 2451 232  Sony 222  Prakash 555  Stamat 111  Stamat = Sony  I believe I can fly! | #Prakash ::: 111, 134, 232, 555, 2451  #Sony ::: 222  #Stamat ::: 222 |

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".



* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

