**Package Installer Exercise Description**

You suddenly have a curious aspiration to create a package installer that can handle dependencies. You want to be able to give the installer a list of packages with dependencies, and have it install the packages in order such that an install won’t fail due to a missing dependency.

This exercise is to write the code that will determine the order of install. Requirements Please complete the exercise in either C# or Javascript. Please use Test Driven Development (TDD) and include your tests. The program should accept an array of strings defining packages and their dependencies. Each string contains the name of a package followed by a colon and space then any dependencies required by that package. For simplicity we’ll assume that a package can have at most one dependency. The program should reject as invalid a dependency specification that contains cycles. The program should output a comma separated string of package names in the order of install, such that a package’s dependency will always precede that package.

*Examples*

VALID INPUT EXAMPLE 1

The input:

[ "KittenService: CamelCaser",

"CamelCaser: " ]

represents two packages, KittenService and CamelCaser, where KittenService depends on CamelCaser. In this case the output should be: "CamelCaser, KittenService" The output indicates that CamelCaser needs to be installed before KittenService.

VALID INPUT EXAMPLE 2

Given the input:

[ "KittenService: ",

"Leetmeme: Cyberportal",

"Cyberportal: Ice",

"CamelCaser: KittenService",

"Fraudstream: Leetmeme",

"Ice: "]

A valid output for the above input would be: "KittenService, Ice, Cyberportal, Leetmeme, CamelCaser, Fraudstream"

INVALID INPUT EXAMPLE

The following input should be rejected because it contains a cycle (Leetmeme -> Cyberportal -> Ice -> Leetmeme): [ "KittenService: ", "Leetmeme: Cyberportal", "Cyberportal: Ice", "CamelCaser: KittenService", "Fraudstream: ", "Ice: Leetmeme" ]