# Problem 2 – SoftUni Course Planning

You are tasked to help planning the next Programing Fundamentals course by keeping track of the lessons, that are going to be included in the course, as well all the exercises for the lessons.

On the first input line you will **receive the initial schedule of lessons and exercises** that are going to be part of the next course, separated by **comma and space ", ".** But before the course starts, there are some changes to be made. Until you receive **"course start"** you will be given **some commands to modify the course schedule**. The possible commands are:

Add:{lessonTitle} – add the lesson to the end of the schedule, **if it does not exist**.

Insert:{lessonTitle}:{index} – insert the lesson to the given index, **if the lesson does not exist and the index is valid.**

Remove:{lessonTitle} – remove the lesson, **if it exists. If the lesson has an Exercise, remove it as well.**

Swap:{lessonTitle}:{lessonTitle} – swap the places of the two lessons, **if they exist. If the lesson has an Exercise, change the Exercise position as well.**

Exercise:{lessonTitle} – add Exercise in the schedule right after the lesson index**, if the lesson exists and there is no exercise already**, in the following format "{lessonTitle}-Exercise". **If the lesson doesn`t exist**, **Add the lesson** in the end of the course schedule**, followed by the Exercise**.

**Each time you Swap or Remove a lesson, you should do the same with the Exercises, if there are any, which follow the lessons.**

## Input / Constraints

* first line – the initial schedule lessons – strings, separated by comma and space ", "
* until **"course start"** you will receive commands in the format described above

## Output

* Print the whole course schedule, each lesson on a new line with its number(index) in the schedule:   
  "{lesson index}.{lessonTitle}"
* Allowed working **time** / **memory**: **100ms** / **16MB**.

## Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| Data Types, Objects, Lists  Add:Databases  Insert:Arrays:0  Remove:Lists  course start | 1.Arrays  2.Data Types  3.Objects  4.Databases | We receive the initial schedule.  Next, we add Databases lesson, because it doesn`t exist.  We Insert at the given index lesson Arrays, because its not present in the schedule.  After receiving the last command and removing lesson Lists, we print the whole schedule. |
| **Input** | **Output** | **Comment** |
| Arrays, Lists, Methods  Swap:Arrays:Methods  Exercise:Databases  Swap:Lists:Databases  Insert:Arrays:0  course start | 1.Methods  2.Databases  3.Databases-Exercise  4.Arrays  5.Lists | We swap the given lessons, because both exist.  After receiving the Exercise command, we see that such lesson doesn`t exist, so we add the lesson at the end, followed by the exercise.  We swap Lists and Databases lessons, the Databases-Exercise is also moved after the Databases lesson.  We skip the next command, because we already have such lesson in our schedule. |

Arrays, Lists, Methods

Swap:Arrays:Methods

course start