**Exercise: Objects and Classes**

Problems for exercise and homework for the ["C# Fundamentals" course @ SoftUni](https://softuni.bg/trainings/2438/csharp-fundamentals-september-2019)  
You can check your solutions in [Judge](https://judge.softuni.bg/Contests/1215)

* **Advertisement Message**

Write a program that **generates random fake advertisement message** to advertise a product. The messages must consist of 4 parts: **phrase** + **event** + **author** + **city**. Use the following predefined parts:

* **Phrases** – {"Excellent product.", "Such a great product.", "I always use that product.", "Best product of its category.", "Exceptional product.", "I can’t live without this product."}
* **Events** – {"Now I feel good.", "I have succeeded with this product.", "Makes miracles. I am happy of the results!", "I cannot believe but now I feel awesome.", "Try it yourself, I am very satisfied.", "I feel great!"}
* **Authors** – {"Diana", "Petya", "Stella", "Elena", "Katya", "Iva", "Annie", "Eva"}
* **Cities** – {"Burgas", "Sofia", "Plovdiv", "Varna", "Ruse"}

The format of the output message is the following: **{phrase} {event} {author} – {city}**.

You will receive the **number of messages** to be generated. Print each random message at a separate line.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 | Such a great product. Now I feel good. Elena – Ruse  Excellent product. Makes miracles. I am happy of the results! Katya – Varna  Best product of its category. That makes miracles. Eva - Sofia |

* **Articles**

Create a **class** **Article** with the following properties:

* **Title** – a string
* **Content** – a string
* **Author** – a string

The class should have a constructor and the following methods:

* **Edit (new content**) – change the old content with the new one
* **ChangeAuthor (new author)** – change the author
* **Rename (new title)** – change the title of the article
* Override the **ToString** method – print the article in the following format:

**"{title} - {content}: {autor}"**

Write a program that reads an article in the following format **"{title}, {content}, {author}"**. On the next line, you will receive a number **n,** representing the number of commands, which will follow after it. On the next **n lines,** you will be receiving the following commands: **"Edit: {new content}"**; **"ChangeAuthor: {new author}"**; **"Rename: {new title}"**. At the end, print the final state of the article.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| some title, some content, some author  3  Edit: better content  ChangeAuthor: better author  Rename: better title | better title - better content: better author |

* **Articles 2.0**

Change the program in such a way, that you will be able to store a **list of articles**. You will not need to use the previous methods any more (**except the ToString method**). On the **first line**, you will receive the number of articles. On the **next lines**, you will receive the **articles in the same format** as in the previous problem: **"{title}, {content}, {author}"**. Finally, you will receive a string: **"title", "content"** or an **"author"**. You need to **order the articles** alphabetically, based on **the** **given** **criteria**.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  Science, planets, Bill  Article, content, Johnny  title | Article - content: Johnny  Science - planets: Bill |
| 3  title1, C, author1  title2, B, author2  title3, A, author3  content | title3 - A: author3  title2 - B: author2  title1 - C: author1 |

* **Students**

Write a program that receives a **count of students - n** and **orders them by grade** in **descending order**. Each student should have a **First name** (string), a **Last name** (string) and a **grade** (a floating-point number).

**Input**

* On the first line, you are going to receive **n - the count of students**
* On the next **n** lines, you will be receiving the info about the students in the following format:

**"{first name} {second name} {grade}"**

**Output**

* Print each student in the following format: **"{first name} {second name}: {grade}"**

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  Lakia Eason 3.90  Prince Messing 5.49  Akiko Segers 4.85  Rocco Erben 6.00 | Rocco Erben: 6.00  Prince Messing: 5.49  Akiko Segers: 4.85  Lakia Eason: 3.90 |