# Lab: Intro to Data Structures

You can check your solutions here: <https://judge.softuni.bg/Contests/3189/Additional-Exercises>.

## Linked Strings

Use LinkedList<T> class and add strings in a **given order**:

* **First string** becomes **first** in the sequence
* **Second string** becomes **last** in the sequence
* **Third string** should be right **after the first one**
* **Fourth string** should be right **before the last one**

**Print** all strings in the right order, separated by **", ".**

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| First  Last  After First  Before Last | First, After First, Before Last, Last |  | how  today  are  you | how, are, you, today |

### Solution

Use AddFirst(), AddLast(), AddAfter() and AddBefore() methods of LinkedList<T>, like shown below:

Text

Description automatically generated

### Submit to Judge

Go to the **folder** with your **solution** and archive the .cs and .csproj files (do not include the bin and obj folders) to a .zip archive:

Graphical user interface, text, application

Description automatically generated

**Submit** the .zip file to Judge as always.

## Bag of Words

Use the class OrderedBag<T> to read n **words** from the console and print the words in a **sorted order** and each on a **new line**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 5  Peter  Maria  Ana  Nina  Mitko | Ana  Maria  Mitko  Nina  Peter |  | 4  apple  banana  pear  watermelon | apple  banana  pear  watermelon |

Note that you should first install SoftUni.Wintellect.PowerCollections from NuGet Packages.

Graphical user interface, application, Word

Description automatically generated

Do not forget to import the namespace Wintellect.PowerCollections at the start of your C# code:



### Solution

Graphical user interface, text, application

Description automatically generated

## Phone Book

Use the **MultiDictionary<K, V>** class to read a **phone book**, where each person can have **multiple phone numbers.**

Note that you should first install SoftUni.Wintellect.PowerCollectionsfrom NuGet Packages.

### Input

The input consists of:

* **N**- number of lines
* **N lines** with the given structure: **"{name} - {phoneNumber1}"**

### Output

Print each **person** with their **phone numbers** the following way: "{name}: {phoneNumber1,phoneNumber2,…}**"**. **Phone numbers** should be separated by **comma** (**","**). Use the default printing of **MultiDictionary<K, V>** class.

Note that **order** in **MultiDictionary<K, V>** can be different (keys order is unpredictable, values keep their insertion order). Sort result by **name** (ascending).

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  Peter - 0877 565 565  Peter - 0875 696 969  Maria - 02 875 5645  Ana - 0877 410 456  Peter - 0879 563 021 | Ana: {0877 410 456}  Maria: {02 875 5645}  Peter: {0877 565 565,0875 696 969,0879 563 021} |

### Hints

You can print the result as shown below, because .Value property formats the result like this: **{phoneNumber1,phoneNumber2,…}.**

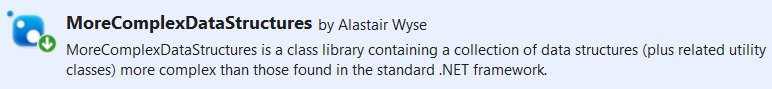
Text

Description automatically generated with low confidence

## Heap of Names

Read n **names** from the console. Use the MaxHeap<T> class to sort names in **descending** **order**. Print each **name**, using the ExtractMax() method.

Note that you should first install MoreComplexDataStructuresfrom NuGet Packages.



### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  Pesho  Kiro  Asen  Miro | Pesho  Miro  Kiro  Asen |

### Hints

**Print** the result with the ExtractMax() method like this:

Text

Description automatically generated with medium confidence