# Lab: Manual String Processing

Problems for exercises and homework for the [“C# Advanced” course @ SoftUni".](https://softuni.bg/trainings/1633/csharp-advanced-may-2017)

Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/595/Manual-String-Processing-Lab>.

## Students Results

Write a program that reads number **N** from the console. After that read **N** lines with students with their results in format **{name} - {firstResult}, {secondResult}, {thirdResult}**

Print **table** on console. Each row must contain:

* **CAdv** - first result, align right with **precision of 2**
* **COOP** - second result, align right with **precision of 2**
* **AdvOOP** - third result, align right with **precision of 2**
* **Average** – average result with **precision of 4**
* Columns must be **separated** with **"|"**
* Don't forget **heading row**

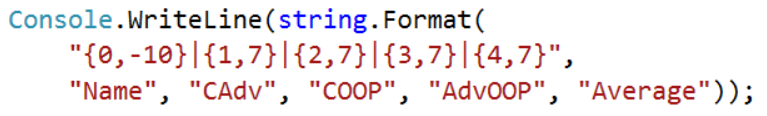
### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  Gosho - 3.33333, 4.4444, 5.555 | Name | CAdv| COOP| AdvOOP|Average|  Gosho | 3,33| 4,44| 5,56| 4,4442| |
| 2  Mara - 5, 4, 3  Gosho - 3, 4, 5 | Name | CAdv| COOP| AdvOOP|Average|  Mara | 5,00| 4,00| 3,00| 4,0000|  Gosho | 3,00| 4,00| 5,00| 4,0000| |

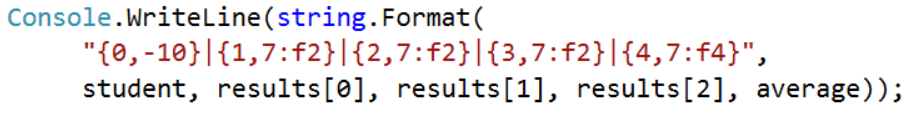
### Hints

It is up to you what type of data structures you will use to solve this problem

First row is **easy**, but long.



Data rows are something terrible:



## Parse URLs

Write a program that parses an URL address given in the format: **[protocol]://[server]/[resource]** and extracts from it the **[protocol]**, **[server]** and **[resource]** elements.

URL is invalid if:

* The protocol separator (**://**) or the resource separator is missing (**/**)
* Contains second time the protocol separator **://**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| <https://softuni.bg/courses/csharp-advanced> | Protocol = https  Server = softuni.bg  Resources = courses/csharp-advance |
| https://www.google.bg/search?q=google&oq=goo&aqs=chrome.0.0j69i60l2://j0j69i57j69i65.2112j0j7&sourceid=chrome&ie=UTF-8 | Invalid URL |

### Hints

In URL standard you can read that **"://"** is used to show where protocol name end. If you have this second time **"://",** itwill be **invalid** URL

Server name ends with **"/"**, but it is **not** part of **resourses**.

Resources use same symbol **"/"** to show when we go deeper in **folders tree**, so be careful.

Think for proper operations over input:

* .Split()
* .Substring()
* .IndexOf()

## Parse Tags

You are given a text. Write a program that changes the text in all regions surrounded by the tags <**upcase**> and </**upcase**> to upper-case.

The tags cannot be nested.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| We are living in a <upcase>yellow submarine</upcase>. We don't have <upcase>anything</upcase> else. | We are living in a YELLOW SUBMARINE. We don't have ANYTHING else. |
| <upcase>StringBuilder</upcase> is <upcase>awesome</upcase> | STRINGBUILDER is AWESOME |

### Hints

Be careful with **replacing** **tags** with **empty** strings.

After replacing tag with empty string, **indexes** in string are **not** the **same**.

## Special Words

Read a string containing a **list of special word**s and a **text** containing some of these words.

Write a program **count special words** in text and **print** their count.

A word separator can be ( ) [ ] < > , - ! ? and space (‘ ’)

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| .NET Microsoft runs framework  .NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. | .NET – 1  Microsoft – 2  runs – 1  framework - 1 |
| .NET Microsoft run  .NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. | .NET - 1  Microsoft - 2  run - 0 |

## Concatenate Strings

Write a program that read number **N** on first line. And on next **N** lines read single words and concatenate them like single line **text** with **" " separator** after each word.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  StringBuilder  is  awesome | StringBuilder is awesome |
| 92  Lorem  ipsum  dolor  sit  amet,  consectetur  adipiscing  elit.  Curabitur  ac  sagittis  purus.  Praesent  rhoncus  tellus  id  felis  dignissim,  vel  gravida  felis  ultricies.  Nunc  ac  mollis  velit.  Pellentesque  tincidunt  ipsum  lectus,  quis  euismod  velit  elementum  malesuada.  Vestibulum  non  tristique  leo,  non  ullamcorper  nunc.  Ut  et  luctus  augue.  Morbi  efficitur  enim  sed  tristique  accumsan.  Sed  ante  augue,  pharetra  quis  leo  vel,  dapibus  pellentesque  lorem.  Vivamus  tincidunt  mauris  odio,  nec  ullamcorper  ipsum  accumsan  id.  Nulla  facilisi.  Sed  id  scelerisque  nibh,  eu  tempor  metus.  Proin  sit  amet  efficitur  ex,  eget  vestibulum  ipsum.  Quisque  malesuada  consequat  semper. | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ac sagittis purus. Praesent rhoncus tellus id felis dignissim, vel gravida felis ultricies. Nunc ac mollis velit. Pellentesque tincidunt ipsum lectus, quis euismod velit elementum malesuada. Vestibulum non tristique leo, non ullamcorper nunc. Ut et luctus augue. Morbi efficitur enim sed tristique accumsan. Sed ante augue, pharetra quis leo vel, dapibus pellentesque lorem. Vivamus tincidunt mauris odio, nec ullamcorper ipsum accumsan id. Nulla facilisi. Sed id scelerisque nibh, eu tempor metus. Proin sit amet efficitur ex, eget vestibulum ipsum. Quisque malesuada consequat semper. |

### Hints

Here we are not only look for **correct** results, but for **time** too.