# Упражнения: Низ и текстови операции

Тествайте решението в Judge: <https://judge.softuni.org/Contests/4163/06-String-Processing-Basics>.

## Валидни потребителски имена

Напишете програма, която чете **имена на потребители**, разделени със **запетая и интервал**. Отпечатайте **валидните** потребителски имена.

Едно потребителско име е **валидно**, когато:

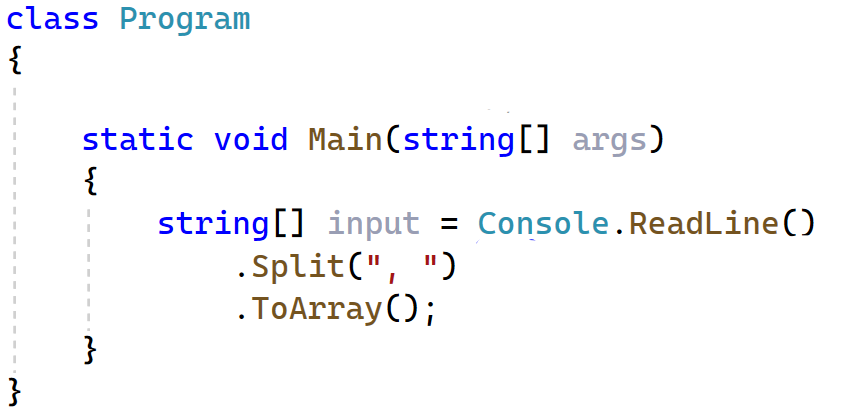
* **Дължината** му е между **3 и 16 символа**
* **Съдържа** само бувки, числа, тирета и долни черти

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| sh, too\_long\_username, !lleg@l ch@rs, jeffbutt | jeffbutt |
| Jeff, john45, ab, cd, peter-ivanov, @smith | Jeff  John45  peter-ivanov |

### Насоки

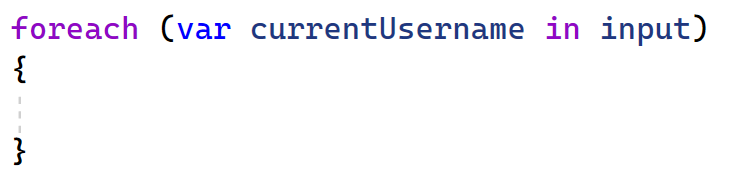
1. Прочетете **имената на потребителите**:



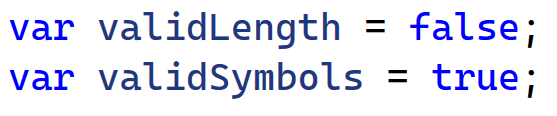
1. Създайте **списък** от **низ**:



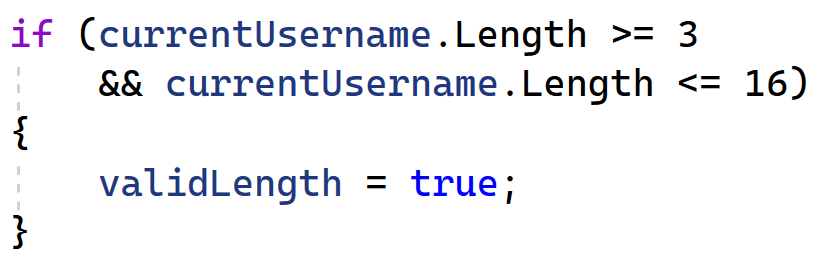
1. Създайте **foreach-цикъл**, който обхожда **имената**, прочетени от входа:



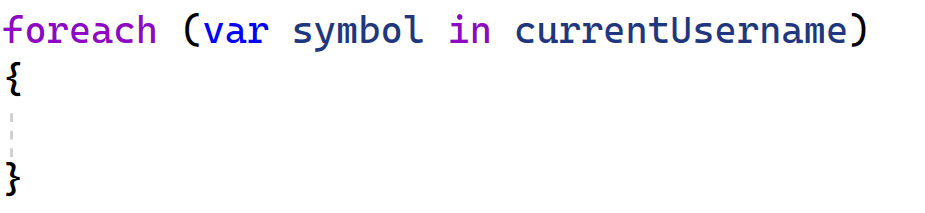
1. В него създайте **две променливи** - **validLength** със стойност **false** и **validSymbols** със стойност **true:**



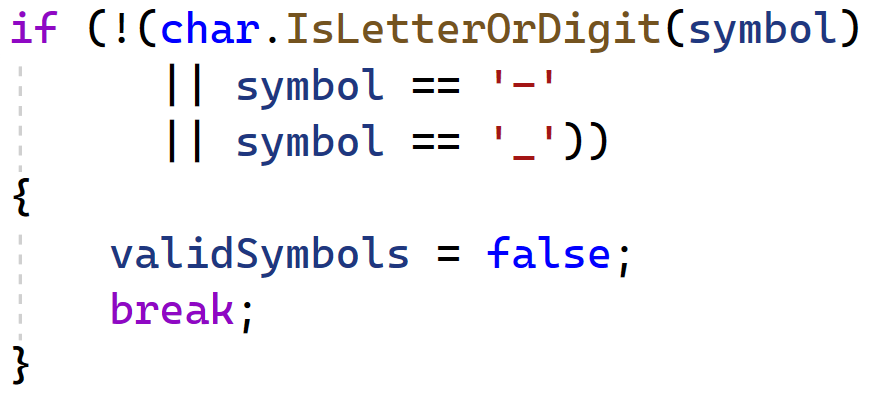
1. Създайте проверка дали **дължината** на **потребителското** име е **по-голяма** или **равна** **3** и **по-малка** или **равна** на **16**:



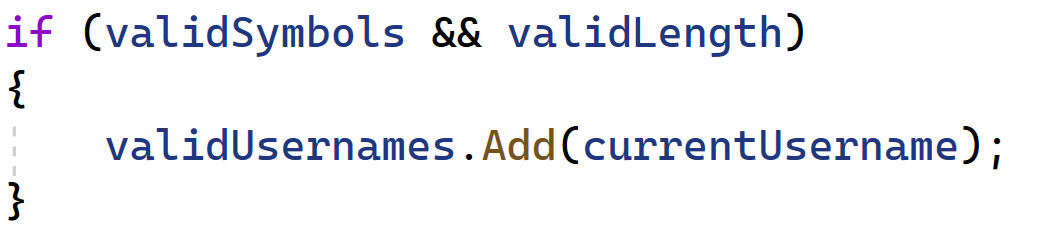
1. След това създайте **foreach-цикъл**, който обхожда символите на **потребителското** **име**:



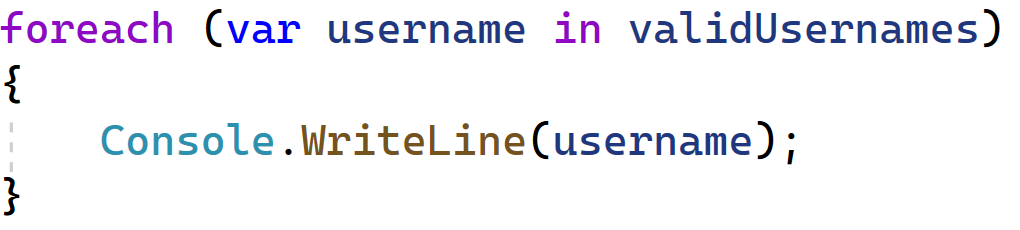
1. В него създайте проверка дали символа **не е число**, **"-"** или **"="**. Ако е така, сложете **стойност** на **validSymbols** **false** и излезте от **foreach-цикъла**:



1. След **foreach-цикълa**,проверете дали **validLength** и **validSymbols** имат стойност **true**. Ако е така, добавете **потребителското име** в **списъка**:



1. Накрая отпечатайте **имената** от **списъка** **validUsernames**:



## Повтаряне на низове

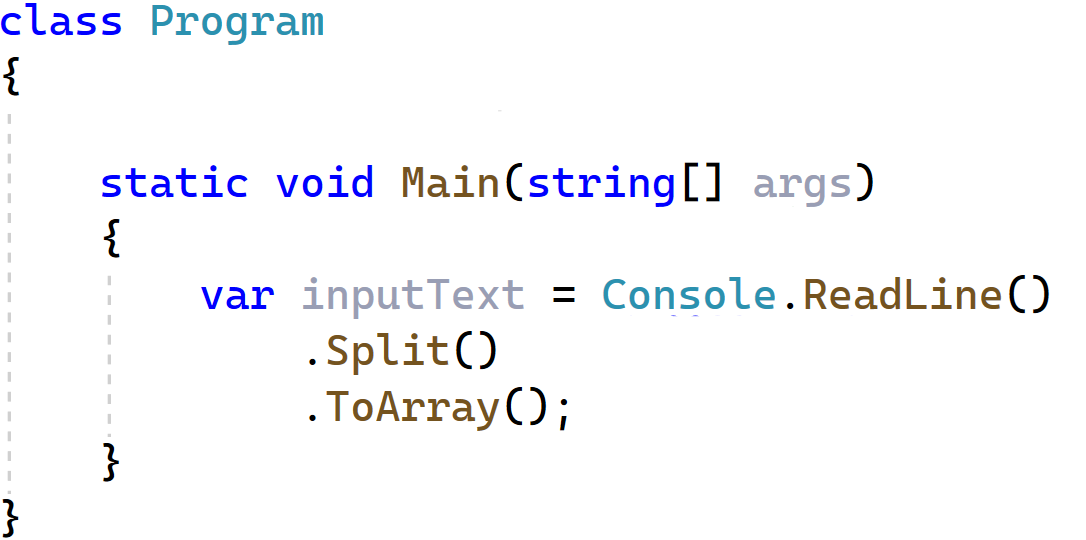
Напишете програма, която чете **масив от низове**. Всеки низ трябва да бъде повторен **N пъти**, където **N** е **дължината** на низа. Накрая ги отпечатайте.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| hi abc add | hihiabcabcabcaddaddadd |
| work | workworkworkwork |
| ball | ballballballball |

### Насоки

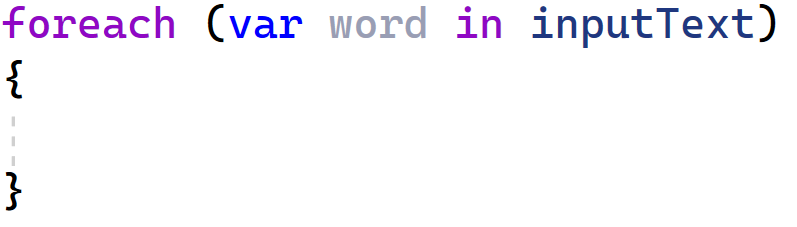
1. Първо прочетете **думите** от **конзолата**:



1. Инициализирайте **StringBuilder**:



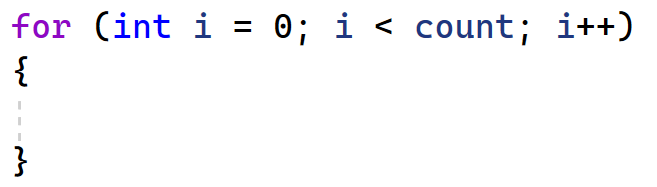
1. **Обходете** всички **низове** на **масива**­:



1. В него създайте **променлива**, която има стойност **дължината на думата**:



1. След това създайте **for-цикъл**, който ще итерира докато променливата **i** е по-малка от **count**:



1. В **for-цикъла** добавете към резултата **думата**:



1. Накрая отпечатайте **резултата**:



## Подниз

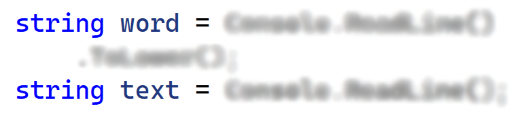
На **първия ред** ще получите **низ**. На **следващия ред** ще получите още един **низ**. Напишете програма, която **изтрива всички** съвпадения от първия във втория низ. Накрая отпечатайте обработения низ.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| Ice  kicegiceiceb | kgb | Първо премахваме ice един път и получаваме "kgiciceeb".  После премахваме ice още веднъж и получаваме "kgiceb".  Накрая премахваме отново и получаваме "kgb" |

### Насоки

1. Първо прочетете **двата** **низа**:



1. **Създайте while-цикъл**, който ще обхожда докато в думата не се среща в текста:



1. Внегоизползвайте метода **Replace(),** за да замените **думата** в **текста**:



1. Накрая отпечатайте текста

## Текст филтър

Напишете програма, която получава **текст** и **забранени думи**. Всички забранени думи трябва да бъдат заминени с "\*" . Забранените думи ще бъдат разделени със **запетая и интервал**.

Всички забранени думи ще бъдат в текста.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Linux, Windows  It is not **Linux**, it is GNU/**Linux**. **Linux** is merely the kernel, while GNU adds the functionality. Therefore we owe it to them by calling the OS GNU/**Linux**! Sincerely, a **Windows** client | It is not \*\*\*\*\*, it is GNU/\*\*\*\*\*. \*\*\*\*\* is merely the kernel, while GNU adds the functionality. Therefore we owe it to them by calling the OS GNU/\*\*\*\*\*! Sincerely, a \*\*\*\*\*\*\* client |

### Насоки

1. Прочетете входа
2. Заменете всички забранени думи с **"\*"**
   * Използвайте метода **Replace()**
   * Използвайте **new string(**char ch, int repeatCount**)**,за да създадете **нов низ**, който съдържа само символ **"\*"** и има дължината на забранената дума

## Файл

Напишете програма, която чете **път до файл**. Вашата задача е да извадите **името на файла** и неговото **разширение**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| C:\Internal\training-internal\Template.pptx | File name: Template  File extension: pptx |
| C:\Projects\Data-Structures\LinkedList.cs | File name: LinkedList  File extension: cs |
| C:\Projects\Data-Structures\README | File name: README  File extension: |
| Hello.txt | File name: Hello  File extension: txt |

## Шифър на Цезар

Напишете програма, която създава **криптиран низ**. Крипитирането работи като **всеки символ** се измества с **три позиции напред**. Например ако имаме **A**, то тя трябва да бъде заменена с **D**, **B** трябва да бъде заменена с **E** и т.н. Накрая отпечатайте **криптирания** **низ**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Programming is cool! | Surjudpplqj#lv#frro$ |
| One year has 365 days. | Rqh#|hdu#kdv#698#gd|v1 |