# Упражнения: Делегати и събития

Можете да проверите решенията си в **Judge системата**: <https://judge.softuni.org/Contests/Practice/Index/4070>

## Сортиране на четни числа

Напишете програма, която чете ред с **цели числа** разделени с **", "**. Отпечатайте **четните числа** от тази поредица,  **сортирани** в **нарастващ** ред.

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 4, 2, 1, 3, 5, 7, 1, 4, 2, 12 | 2, 2, 4, 4, 12 | 1, 3, 5 |  | 2, 4, 6 | 2, 4, 6 |

### Насоки

Можете сами да изберете типа данни, които ще използвате, за да решите задачата. Използвайте **функционален филтър с делегат (или ламбда функция)** и сортирайте колекцията от числа.

## Сума от числа

Напишете програма, която чете ред с **цели числа**, разделени с **", "**. Отпечатайте на два реда **броя** на числата и тяхната **сума**. Използвайте вградената **agregate** функция за сумиране.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4, 2, 1, 3, 5, 7, 1, 4, 2, 12 | 10  41 |
| 2, 4, 6 | 3  12 |

## Брой на думите с главни букви

Напишете програма, която чете ред с **текст** от конзолата. Отпечатайте **всички думи**, които **започват с главна буква** в същия ред, в който сте ги получили.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| The following example shows how to use Function | The  Function |
| Write a program that reads one line of text from console. Print count of words that start with Uppercase, after that print all those words in the same order like you find them in text. | Write  Print  Uppercase, |

### Насоки

Използвайте **Func<string,** **bool>** за филтриране на входните думи и използвайте **" "**, за да разделите думите.

## Добавяне на ДДС

Напишете програма, която чете един ред от цени от тип **double**, разделени с **", "**. Отпечатайте **цените** с добавен **ДДС** за всяка цена. **ДДС** е равно на 20% от цената. **Форматирайте** цените до **2 знака** след десетичната запетая. **Подредбата** им трябва да остане **същата**. Използвайте функция **double 🡪 double**, за да изчислите ВАТ.

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| 1.38, 2.56, 4.4 | 1.66  3.07  5.28 | 1, 3, 5, 7 | 1.20  3.60  6.00  8.40 |

## Филтриране по възраст

Напишете програма, която получава цяло число **N** на първия ред. На следващите **N** реда прочетете двойки от **"[име], [възраст]"**. След това прочетете 3 реда с:

* **Condition** – "**younger**" or "**older**"
* **Age** – Integer
* **Format** – "**name**", "**age**" or "**name** **age**"

В зависимост от **условието**, отпечатайте правилните **двойки** в правилния **формат**. **Не използвайте** вградена функционалност от .NET. **Създайте свои методи**.

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  older  20  name age | Lucas - 20  Mia - 29  Noah - 31 | 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  younger  20  name | Tomas  Simo |  | 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  younger  50  age | 20  18  29  31  16 |

## Действие за отпечатване

Напишете програма, която прочита колекция от **стрингове** от конзолата и след това **ги отпечатва** на конзолата. Всяко име трябва да се отпечатва на **нов ред**. Използвайте **Action<T>**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Lucas Noah Tea | Lucas  Noah  Tea |

## Рицари

Напишете програма, която прочита колекция с **имена** (стрингове) от конзолата, добавя "**Sir**" пред всяко име и ги **отпечатва** на конзолата. Използвайте **Action<T>**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Eathan Lucas Noah StanleyRoyce | Sir Eathan  Sir Lucas  Sir Noah  Sir StanleyRoyce |

## Min функция

Напишете програма, която **прочита** от конзолата поредица от **цели числа** и **отпечатва** **най-малкото число** в колекцията. Използвайте **Func<T, T>**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 4 3 2 1 7 13 | 1 |

## Четни и нечетни числа

Зададени са долна и горна граница за диапазон от **цели числа**. След това е дадена команда, която уточнява дали да филтрирате **само четните** или **само нечетните** числа в дадения диапазон. Използвайте **Predicate<T>**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 10  odd | 1 3 5 7 9 |
| 20 30  even | 20 22 24 26 28 30 |

**TODO**: да се измисли задача с EventHandler, например нещо с Timer