# Упражнения: Делегати и събития

Можете да проверите решенията си в **Judge системата**: <https://judge.softuni.bg/Contests/3168/Delegates-and-Events>

## 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** 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 ВАТ

Напишете програма, която чете един ред от цени от тип **double**, разделени с **", "**. Отпечатайте **цените** с добавен **ВАТ** за всяка цена. **ВАТ** е равен 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 |

## Обърнете и изключете

Напишете програма, която **обръща** колекция от **цели числа** и **премахва елементите**, които се делят на дадено цяло число **n**. Използвайте **предикати/функции**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 4 5 6  2 | 5 3 1 |
| 20 10 40 30 60 50  3 | 50 40 10 20 |

## Предикат за имена

Напишете програма, която филтрира **списък с имена** в зависимост от тяхната **дължина**. На първия ред ще получите **цяло число n**, което представлява **дължината на името**. На втория ред ще получите имена, разделени с интервал. Напишете функция, която отпечатва **само имената**, чиято **дължина** е **по-малка или равна на n**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4  Nathaniel Molly Dan Joe Glen | Dan  Joe  Glen |
| 4  Alexia Cara Robin Lynda | Cara |

## Функция за сравнение

Напишете сравняваща функция, която сортира **всички четни числа** преди **всички нечетни** числа в **нарастващ ред**. Подайте я на функцията **Array.Sort()** и **отпечатайте** резултата.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 4 5 6 | 2 4 6 1 3 5 |
| -3 2 | 2 -3 |

## Списък с предикати

Намерете всички числа в диапазона **1...N**, които се **делят** на числата от **дадена поредица**. На първия ред ще получите **цяло число N**, което е **края на диапазона**. На втория ред ще получите поредица от цели числа, които са **делителите**. Използвайте предикати/функции.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 10  1 1 1 2 | 2 4 6 8 10 |
| 100  2 5 10 20 | 20 40 60 80 100 |

## Парти от предикати!

Родителите на Карлос са на ваканция за празниците и той планира парти вкъщи. За съжаление, организационните му способности са слаби, така че вие имате задачата да му помогнете с резервациите.

На първия ред ще получите **списък** с **всички имена** на хората, които ще дойдат. На следващите редове, до получаване на команда **"Party!"**, ще трябва да **удвоите** или да **премахнете всички хора**, които отговарят на даден **критерий**. Има **три различни критерия**:

* Имената трябва да **започват** с **дадения стринг**
* Имената трябва да **завършват** с **дадения стринг**
* Имената трябва да имат **дадената дължина**

Накрая **отпечатайте всички гости**, които ще дойдат на партито, разделени с "," и след това **добавете** към края на стринга **ending** "are going to the party!". Ако **няма гости** на партито, отпечатайте "Nobody is going to the party!".

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Paul Alice Hector  Remove StartsWith P  Double Length 5  Party! | Alice, Alice, Hector are going to the party! |
| Peter  Double StartsWith Pete  Double EndsWith eter  Party! | Peter, Peter, Peter, Peter are going to the party! |
| Philip  Remove StartsWith P  Party! | Nobody is going to the party! |

## Модул за филтър за парти

Имплементирайте филтриращ модул за партито. Първо, на този филтър, който ще наречем PRFM (Party Reservation Filter Module) , се **подава списък с покани**. След това PRFM получава **поредица от команди**, които конкретизират дали да се добави или премахне даден филтър.

Всяка PRFM команда има следния формат:

"**{command;filter type;filter parameter}"**

Можете да получите следните PRFM команди:

* "**Add filter**"
* "**Remove filter**"
* "**Print**"

Възможните типове PRFM филтри са:

* "**Starts with**"
* "**Ends with**"
* "**Length**"
* "**Contains**"

Всички параметри за PRFM филтрите са стрингове (или цяло число за филтъра "**Length"**). Всяка команда ще бъде валидна т.е. няма нужда да проверявате филтрите. Входът ще **приключи** с команда "**Print**", след което трябва да отпечатате **всички посетители** на партито, останали **след филтрирането**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Paul Mark Sandy  Add filter;Starts with;P  Add filter;Starts with;M  Print | Sandy |
| Peter Miles Jorge  Add filter;Starts with;P  Add filter;Starts with;M  Remove filter;Starts with;M  Print | Miles Jorge |

## Калкулатор за събития

Напишете програма, която събира две числа. Ще получите **числата**, разделени с **интервал**. Отпечатайте **сумата** от числата и ако тя е **нечетно число**, активирайте **събитие**, което отпечатва съобщението "\*\*\*\*\*\*\*\*Event Executed : This is Odd Number\*\*\*\*\*\*\*\*\*\*", използвайки **делегати**.

Създайте **публичен** клас AddTwoNumbers, декларирайте в него делегат, събитие и създайте метод Add(). Методът трябва да пресмята сумата и да провери дали **събитието е активирано**. В класа StartUp създайте обект от класа AddTwoNumbers, абонирайте се към събитието и използвайте метода Add().

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3 4 | 7  \*\*\*\*\*\*\*\*Event Executed : This is Odd Number\*\*\*\*\*\*\*\*\*\* |
| 2 2 | 4 |

### Насоки

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

## Събитие при натискане на бутон

Важно: Тази задача не може да бъде тествана в Judge!

Напишете програма, която **активира събитие**, когато се натисне бутонът [a] или [b] на клавиатурата. Използвайте метод, за да напишете цвета на конзолата със следното съобщение: "You pressed the 'A' key.", "You pressed the 'B' key." or "No event handler for key {key}".

За да го направите, създайте **публичен** клас Keyboard, който съдържа следните членове и методи:

* Един **делегат** PressKeyEvent
* Две **събития**, равни на null, например PressKeyA и PressKeyB
* void PressKeyAEvent() – ако събитието PressKeyA не е null, извикайте го с базовия системен метод .Invoke()
* void PressKeyBEvent() – ако събитието PressKeyB не е null, извикайте го с базовия системен метод .Invoke()
* void Start() – **чете** от конзолата, **използва switch** в зависимост дали е натиснат бутонът [a] и [b] и извиква съответния метод. **Поведението по подразбиране** (ако е натиснат друг бутон) трябва да е да се изпише на конзолата "No event handler for key {key}"

Създайте следните методи в главния метод, които можете да подавате като аргументи на делегата:

Graphical user interface, text, application

Description automatically generated

Използвайте метода start, за да стартирате програмата.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| a | You pressed the 'A' key. |
| b | You pressed the 'B' key. |
| f | No event handler for key f. |

### Насоки

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated