# Упражнения: Сетове и речници

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/3175/Sets-Dictionaries>.

## **Брой на еднакви стойности в масив**

Напишете програма, която намира колко пъти едно число се среща в масив от реални числа.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Пример** |
| -2.5 4 3 -2.5 -5.5 4 3 3 -2.5 3 | -2.5 - 3 times  4 - 2 times  3 - 4 times  -5.5 - 1 times |
| 2 4 4 5 5 2 3 3 4 4 3 3 4 3 5 3 2 5 4 3 | 2 - 3 times  4 - 6 times  5 - 4 times  3 - 7 times |

## **Средноаретметичен успех на ученици**

Напишете програма, която чете **имената на ученици** с техните **оценки** и ги добавя към **бележниците им.** Накрая отпечатайте **името на всеки** **ученик** с неговите оценки и **средноаритметичен успех**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 7  Ivancho 5.20  Mariika 5.50  Ivancho 3.20  Mariika 2.50  Stamat 2.00  Mariika 3.46  Stamat 3.00 | Ivancho -> 5.20 3.20 (avg: 4.20)  Mariika -> 5.50 2.50 3.46 (avg: 3.82)  Stamat -> 2.00 3.00 (avg: 2.50) |
| 4  Vladimir 4.50  Petko 3.00  Vladimir 5.00  Petko 3.66 | Vladimir -> 4.50 5.00 (avg: 4.75)  Petko -> 3.00 3.66 (avg: 3.33) |

|  |  |
| --- | --- |
| 5  Gosho 6.00  Gosho 5.50  Gosho 6.00  Ivan 4.40  Petko 3.30 | Gosho -> 6.00 5.50 6.00 (avg: 5.83)  Ivan -> 4.40 (avg: 4.40)  Petko -> 3.30 (avg: 3.30) |

### Насоки

* Използвайте **речник** (string à List<decimal>)
* Проверете дали името съществува. Ако липсва, го **добавете** в речника.
* Преминете през всички ключ-стойност двойки в речника за да отпечатате резултата.
* За да пресметнете по-бързо средноаритметичен успех, може да използвате .Average().

## Хранителен магазин

Напишете програма, която **принтира и събира** информация за **хранителните магазини** в София. Докато не получите командата "Revision", ще полочвате команди във формата: "{магазин}, {продукт}, {цена}".Имайте предвид, че ако получите магазин, който съществува, просто добавете информацията за продукта.

Трябва да ги отпечатате подредени по име:

{Хранителен магазин}->

Product: {Продукт}, Price: {Цена}

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| lidl, juice, 2.30  fantastico, apple, 1.20  kaufland, banana, 1.10  fantastico, grape, 2.20  Revision | fantastico->  Product: apple, Price: 1.2  Product: grape, Price: 2.2  kaufland->  Product: banana, Price: 1.1  lidl->  Product: juice, Price: 2.3 |
| tmarket, peanuts, 2.20  GoGrill, meatballs, 3.30  GoGrill, HotDog, 1.40  tmarket, sweets, 2.20  Revision | GoGrill->  Product: meatballs, Price: 3.3  Product: HotDog, Price: 1.4  tmarket->  Product: peanuts, Price: 2.2  Product: sweets, Price: 2.2 |

## Градове по континент и държава

Напишете програма, която чете континенти, държави и градове. Сложете ги в сложен речник и ги отпечатайте.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 9  Europe Bulgaria Sofia  Asia China Beijing  Asia Japan Tokyo  Europe Poland Warsaw  Europe Germany Berlin  Europe Poland Poznan  Europe Bulgaria Plovdiv  Africa Nigeria Abuja  Asia China Shanghai | Europe:  Bulgaria -> Sofia, Plovdiv  Poland -> Warsaw, Poznan  Germany -> Berlin  Asia:  China -> Beijing, Shanghai  Japan -> Tokyo  Africa:  Nigeria -> Abuja |
| 3  Europe Germany Berlin  Europe Bulgaria Varna  Africa Egypt Cairo | Europe:  Germany -> Berlin  Bulgaria -> Varna  Africa:  Egypt -> Cairo |
| 8  Africa Somalia Mogadishu  Asia India Mumbai  Asia India Delhi  Europe France Paris  Asia India Nagpur  Europe Germany Hamburg  Europe Poland Gdansk  Europe Germany Danzig | Africa:  Somalia -> Mogadishu  Asia:  India -> Mumbai, Delhi, Nagpur  Europe:  France -> Paris  Germany -> Hamburg, Danzig  Poland -> Gdansk |

### Насоки

* Използвайте **сложен речник** (string à (Dictionary à List<string>))
* Проверете дали континента съществува в речника. Ако липсва, добавете го.
* Проверете дали държавата съществува в речника. Ако липсва, добавете я.
* Преминете през всички **ключ-стойност** в речника и отпечатайте резултата.

## Уникални имена

Напишете програма, която чете **списък от имена** и отпечатва **уникалните** имена.

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 8  Ivan  Pesho  Ivan  Stamat  Pesho  Alice  Peter  Pesho | Ivan  Pesho  Stamat  Alice  Peter |  | 7  Lyle  Bruce  Alice  Easton  Shawn  Alice  Shawn  Peter | Lyle  Bruce  Alice  Easton  Shawn |  | 6  Roki  Roki  Roki Roki  Roki  Roki | Roki |

### Насоки

Сложете имената в HashSet<string>, за да вземете уникалните имена.

## Броя на еднакви символи в един низ

**Напишете програма, която намира броя на буквите в низ с изключение на (' ')**. Отпечатайте резултата в следния формат: **{буква} -> {брой пъти срещания}**

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| text | t -> 2  e -> 1  x -> 1 | text text text | t -> 6  e -> 3  x -> 3 |

## Сет от елементи

Напишете програма, която отпечатва **всички елементи на сет**. На първия ред ще получите две числа - **n** и **m**, които представляват дължината на два сета. На следващите **n + m** реда ще получите **n** числа за **първия** сет, а след това **m** числа за **втория** сет. Накрая отпечатайте **числата**, които се появяват и в **двете колекции**. Трябва да бъдат в **реда от първи сет**.

**Например:**

* Сет с дължина n = 4: {1, **3**, **5**, 7}
* Сет с дължина m = 3: {**3**, 4, **5**}
* Елементите от **двата сета са** -> {**3**, **5**}

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 4 3  1  3  5  7  3  4  5 | 3 5 | 2 2  1  3  1  5 | 1 |

## -------- TODO: MOVE TO ADVANCED

## Периодична таблица

Напишете програма, която съдържа **уникални елементи**. На първия ред ще получите числото **n**. На следващите **n реда** ще получите химични съединения, разделени с **интервал**. Вашата задача е да отпечатате **уникалните елементи** в **азбучен ред**:

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 4  Ce O  Mo O Ce  Ee  Mo | Ce Ee Mo O | 3  Ge Ch O Ne  Nb Mo Tc  O Ne | Ch Ge Mo Nb Ne O Tc |

## Четни брой пъти

Напишете програма, която **отпечатва число**, което се повтаря **четни пъти**. На първия ред ще получите **n** - броя на числата, които ще получите на следващите редове. Гарантирано е, че само **едно число** ще се появи **четен брой пъти**. Вашата задача е да го **намерите** и да го **отпечатате**.

### Примери

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

## Легендарни продукти

Свърши ли сте цялата работа и единственото, което ви е останало е да вземете легендарния **продукт**. Но за жалост трябва да вземете продукта с малко фармене. Нямате притенции и затова ще вземете продукта само, за който имате пари. Възможните **продукти** са:

* **Shadowmourne** - трябва да имате **250 Shards**;
* **Valanyr** - трябва да имате **250 Fragments**;
* **Dragonwrath** - трябва да имате **250 Motes**;

**Shards, Fragments** и **Motes** са материали и всичко останало са боклуци. Ще получите редове за вход във формата:

**2 motes 3 ores 15 stones**

Програмата спира, когато **Fragments, Motes** или **Shards** достигне количество над 250. След това трябва да отпечатате материалите, сортирани по **количество във възходящ ред**, после по имена в **низходящ ред**, отпечатани на **нов ред**. Накрая отпечатайте боклуците сортирани по **азбучен ред**.

### Вход

* Всеки ред ще бъде съставен от: **{количество} {материал/боклук} {количество} {материал/боклук} … {количество} {материал/боклук}**

### Изход

* На първият ред отпечатайте продукта: **{Продукт} obtained!**
* На следващите три реда отпечатайте материалите подредени по количество във **възходящ ред**
  + Ако има **два** материала с еднакво количество, трябва да ги сортирате по **низходящ ред**
* Накрая отпечатайте боклуците сортирани по **азбучен ред**
  + Всички боклуци трябва да бъдат във формата: **{боклук}: {количество}**
  + Имената трябва да бъдат с **малки букви**, освен първата буква

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3 Motes 5 stones 5 Shards  6 leathers 255 fragments 7 Shards | Valanyr obtained!  fragments: 5  shards: 5  motes: 3  leathers: 6  stones: 5 |
| 123 silver 6 shards 8 shards 5 motes  9 fangs 75 motes 103 MOTES 8 Shards  86 Motes 7 stones 19 silver | Dragonwrath obtained!  shards: 22  motes: 19  fragments: 0  fangs: 9  silver: 123 |

## Продукти

Напишете програма, която пази **информация за продуктите** и тяхната **цена**. Всеки продукт има **име, цена** и **количество**. Ако продукта **не съществува**, трябва да го **добавете** с неговото **количество**.

Ако получите продукт, който **съществува**, **увеличете** неговото количество . Ако цената е различна, я променете.

Ще получите **името**, **цената** и **количеството** на продукта на **нов ред**. Докато не получите командата "**buy**", продължавайте да добавяте продукти.

### Вход

* Докато не получите командата "**buy**", продуктите ще бъдат в следния формат: "**{име} {пари} {качество}**"
* Данните на продукта винаги ще бъдат **разделени с интервал**.

### Изход

* Отпечатайте информацията за всеки продукт в следният формат:  
  **"{Име на продукта} -> {Общата цена}"**
* Закръглете числата до **втория знак след десетичната запетая**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Beer 2.20 100  IceTea 1.50 50  NukaCola 3.30 80  Water 1.00 500  buy | Beer -> 220.00  IceTea -> 75.00  NukaCola -> 264.00  Water -> 500.00 |
| Beer 2.40 350  Water 1.25 200  IceTea 5.20 100  Beer 1.20 200  IceTea 0.50 120  buy | Beer -> 660.00  Water -> 250.00  IceTea -> 110.00 |
| CesarSalad 10.20 25  SuperEnergy 0.80 400  Beer 1.35 350  IceCream 1.50 25  buy | CesarSalad -> 255.00  SuperEnergy -> 320.00  Beer -> 472.50  IceCream -> 37.50 |

## СофтУни паркинг

СофтУни току що са получи новият си паркинг. Толкова е нов, че даже си има онлайн валидация за паркинг. С изключение, че онлайн паркинга не работи. Само получава данни за потребителя, но не знае какво да прави с тях. Добре че сте в екипа от програмисти и знаете как да го управите, нали?

Напишете програма, която валидира място за паркиране в онлайн услугата. Потребителя може да се регистрира в паркинга или да излезе.

Програмата ще **получи 2 команди**:

* "register {потребител} {регистрационен номер на колата}":
  + Системата подържа само една кола за един човек. Ако потребител се опита да добави още един регистрационен номер на кола, отпечатайте:  
    "ERROR: already registered with plate number {регистрационен номер на колата}"
  + Ако проверката по-горе премине успешно, отпечатайте:  
     **"{**потребител**} registered {**регистрационен номер на колата} **successfully"**
* "unregister {потребител}":
  + Ако няма такъв потребител, отпечатайте:  
    "ERROR: user {username} not found"
  + В противен случай, отпечатайте:  
    "{username} unregistered successfully"

Преди да свърши програмата, трябва да отпечатате **потребителите** и **техните номера** в следния формат:

"**{**потребител**} => {**регистрационен номер на колата**}**"

### Вход

* На първия ред ще получите **n** - **броя на командите**
* На следващите n редове ще получите команди. Възможните команди са:
  + Регистряция: "register {потребител} {регистрационен номер на колата}"
  + Излизане: "unregister {потребител}"

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5  register John CS1234JS  register George JAVA123S  register Andy AB4142CD  register Jesica VR1223EE  unregister Andy | John registered CS1234JS successfully  George registered JAVA123S successfully  Andy registered AB4142CD successfully  Jesica registered VR1223EE successfully  Andy unregistered successfully  John => CS1234JS  George => JAVA123S  Jesica => VR1223EE |
| 4  register Jony AA4132BB  register Jony AA4132BB  register Linda AA9999BB  unregister Jony | Jony registered AA4132BB successfully  ERROR: already registered with plate number AA4132BB  Linda registered AA9999BB successfully  Jony unregistered successfully  Linda => AA9999BB |
| 6  register Jacob MM1111XX  register Anthony AB1111XX  unregister Jacob  register Joshua DD1111XX  unregister Lily  register Samantha AA9999BB | Jacob registered MM1111XX successfully  Anthony registered AB1111XX successfully  Jacob unregistered successfully  Joshua registered DD1111XX successfully  ERROR: user Lily not found  Samantha registered AA9999BB successfully  Joshua => DD1111XX  Anthony => AB1111XX  Samantha => AA9999BB |

## ------ TODO: MOVE TO MORE EXERCISES

## Курсове

Напишете програма, която пази информация за **курсове**. Всеки курс съдържа имена на студенти.

Ще получите **име на курса** и **имена на студенти**, докато не получите "**end**". **Ако няма такъв курс**, го добавете. След това регистрирайте студента. Когато получите командата "**end**", отпечатайте **имената на курсовете** с **имената на студентите**. Подредете ги по брой студенти **в низходящ ред**. Студентите трябва да бъдат **сортирани по азбучен ред**.

### Вход

* Докато не получите командата "**end**", ще получавате вход в следния формат:

"**{Име на курса} : {Име на студента}**"

* Винаги ще бъдат разделени с **" : ".**

### Изход

* Отпечатайте информацията за **всеки кур**с в следния формат:
* **"{Име на курса}: {Броя на студентите в него}"**
* Отпечатайте информацията за **всеки студент** в следния формат:  
  **"-- {Име на студнет}"**

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Programming Fundamentals : John Smith  Programming Fundamentals : Linda Johnson  JS Core : Will Wilson  Java Advanced : Harrison White  end | Programming Fundamentals: 2  -- John Smith  -- Linda Johnson  JS Core: 1  -- Will Wilson  Java Advanced: 1  -- Harrison White |
| Algorithms : Jay Moore  Programming Basics : Martin Taylor  Python Fundamentals : John Anderson  Python Fundamentals : Andrew Robinson  Algorithms : Bob Jackson  Python Fundamentals : Clark Lewis  end | Python Fundamentals: 3  -- Andrew Robinson  -- Clark Lewis  -- John Anderson  Algorithms: 2  -- Bob Jackson  -- Jay Moore  Programming Basics: 1  -- Martin Taylor |

## Компания от служители

Напишете програма, която пази информация за компании и техните служители.

Ще получите име на компания и лични карти на работниците, докато не получите командата "**End**". Ще добавяте работник за всяка компания. Имайте на предвид, че не можете да имате двама работника с еднаква лична карта.

Когато приключите, сортирайте имената на компаниите по азбучен ред.

Отпечатайте имената на компаниите и техните работници в следния формат:

1. **{Име на компания}**
2. **-- {лична карта на работника1}**
3. **-- {лична карта на работника2}**
4. **-- {лична карта на работникаN}**

### Вход

* Докато получите "**End**", ще четете команди в следния формат: "**{Име на компанията} -> {лична карта на работника}**"

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| SoftUni -> AA12345  SoftUni -> BB12345  Microsoft -> CC12345  HP -> BB12345  End | HP  -- BB12345  Microsoft  -- CC12345  SoftUni  -- AA12345  -- BB12345 | SoftUni -> AA12345  SoftUni -> CC12344  Lenovo -> XX23456  SoftUni -> AA12345  Movement -> DD11111  End | Lenovo  -- XX23456  Movement  -- DD11111  SoftUni  -- AA12345  -- CC12344 |

## \* ForceBook

Потребителите не могат да запомнят от коя страна се намират. Затова ви се поставя задача за управление на профилите им. Програмата трябва да съхранява от коя страна се намира даден потребител.

Ще получите няколко входа, които могат да бъдат:

{страна} | {потребител}

{потребител} -> {страна}

**Потребителят** и **страната** са низове.

Ако получите {страна} | {потребител}, трябва да проверите дали съществува такъв потребител. Ако няма, добавете го.

Ако получите {потребител} -> {страна}, трябва да проверите дали потребителя съществува. Ако е така, трябва да му промените страната. Ако няма такъв потребител, добавете го до в страната.  
След това трябва да отпечатате: **"{Потребителя} joins the {страната} side!"**

Програмата спира, когато получите "Lumpawaroo". След това трябва да отпечатате страните сортиране по броя на потребители във възходящ ред и след това ги сортирайте по имена. Потребителите трябва да бъдат сортирани по имена.

### Вход

* Входа ще бъде във формата по-горе
* Когато получите "Lumpawaroo", програмата спира да чете.

### Изход

* Изходът за всяка страна трябва да бъде **сортиран по брой на потребители** и **след това по име**. Потребителите трябва да бъдат **сортирани по име**.
* Изходът трябва да бъде във формата:

Side: {страна}, Members: {брой потребител}

! {потребител}

! {потребител}

! {потребител}

* В случай, че нямате **потребители в определена** **страна**, **не трябва** да отпечатвате информация за страната.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| Light | Gosho  Dark | Pesho  Lumpawaroo | Side: Dark, Members: 1  ! Pesho  Side: Light, Members: 1  ! Gosho | Регистрираме Gosho в страната Light и Pesho в страната Dark. След като получим "Lumpawaroo", отпечатваме и двете страни сортирани по брой на потребителите и по име. |
| Lighter | Royal  Darker | DCay  Ivan Ivanov -> Lighter  DCay -> Lighter  Lumpawaroo | Ivan Ivanov joins the Lighter side!  DCay joins the Lighter side!  Side: Lighter, Members: 3  ! DCay  ! Ivan Ivanov  ! Royal | Ivan Ivanov няма профил и затова го регистрираме като го добавяме в страната Lighter.  Премахваме **DCay** от Darker и го добавяме в страната Lighter.  Отпечатваме страната Lighter, защото само тя има потребители |

## \*Резултати от изпита в СофтУни

Статистиката на системата Judge от изпита на Programing Fundamentals не работи правилно. Трябва да напишете програма, която приема всички решения и ги анализира. Трябва да ги събирате и да отпечатате финалния резултат и статистиката за всеки език.

Докато не получите "exam finished", ще получавате вход в следния формат: "{име на потребител}-{език}-{точки}". Трябва да съхраните потребителското име, точките и езика на студента.

Ще получавате също команда за блокиране на потребител, защото е преписвал, в следния формат: "{име на потребител}-banned". В такъв случай трябва да го премахнете от състезанието, но да запазите неговите опити.

След като получите "exam finished" ,отпечатайте всички участници, сортирани по низходящ ред по точки, след това по потребителско име, в следния формат:

Results:

{име на потребител} | {точки}

…

След това отпечатайте всеки език като ги сортирате по броя на резултатите по низходящ ред и след това по името на език. Трябва да бъдат отпечатани по следния формат:

Submissions:

{език} – {брой решения за определен език}

…

### Вход

Докато не получите **"exam finished"** ще получавате решения на участниците в следния формат:  
"{потребителско име}-{език}-{точки}".

Ще получавате команди за блокиране -> "{име на потребител}-banned"

### Изход

* Отпечатайте
* Отпечатайте резултатите от изпита за всеки участник, подредени в низходящ ред по максимални точки и след това по потребителско име, в следния формат:

Results:

{username} | {points}

…

* Езиците трябва да бъдат сортирани по възходящ ред по брой опити и след това по име на езика. Изходът трябва да бъде следния формат:

Submissions:

{Език} – {брой опити}

…

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| Pesho-Java-84  Gosho-C#-84  Gosho-C#-70  Kiro-C#-94  exam finished | Results:  Kiro | 94  Gosho | 84  Pesho | 84  Submissions:  C# - 3  Java - 1 | Сортираме участниците по възходящ по максимален брой точки, по техните имена. Отпечатваме само потребителското име и максималния брой точки.  Сортираме езиците по техния брой опити във възходящ ред и по име на езика |
| Pesho-Java-91  Gosho-C#-84  Kiro-Java-90  Kiro-C#-50  Kiro-banned  exam finished | Results:  Pesho | 91  Gosho | 84  Submissions:  C# - 2  Java - 2 | Киро е блокиран и затова е премахнат от съзтезанието, но неговите опити остават в брой опити за езиците.  Затова пише, че са участвали само 2 участника вместо 4. |

## \* Гардероб

Напишете програма, която ви помага да изберете какви дрехи да носите от вашия **гардероб**. Ще получите **дрехи**, които са във вашият гардероб. Сортирани по **цвят** във формата:

"{цвят} -> {дреха1},{дреха2},{дреха3}…"

Ако получите определен цвят, който вече съществува, просто добавете **дрехата**. Може също да получавате повтарящи се **дрехи** с **определен цвят**. Трябва да пазите и тяхната **бройка**.

Накрая ще получите **цвят и дреха**, която трябва да **намерите** в гардероба и да отпечатате всички продукти и тяхната бройка за всеки цвят във формата:

**"{Цвят} {дреха}"**

* **"{цвят} clothes:**
* **\* {дреха1} - {брой}**
* **\* {дреха2} - {брой}**
* **\* {дреха3} - {брой}**
* **…**
* **\* {дрехаN} - {брой}"**
* **Ако откриете дрехата трябва да отпечатате "(found!)" до нея**
* **"\* {дрехаN} - {брой} (found!)"**

### Вход

* На **първия ред** ще получите числото **n** - броя на редовете, който ще получите.
* На **следващите n реда** ще получите **дрехи** във формата по-горе.

### Изход

* Отпечатайте дрехи от **вашия гардероб** във формата по-горе.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4  Blue -> dress,jeans,hat  Gold -> dress,t-shirt,boxers  White -> briefs,tanktop  Blue -> gloves  Blue dress | Blue clothes:  \* dress - 1 (found!)  \* jeans - 1  \* hat - 1  \* gloves - 1  Gold clothes:  \* dress - 1  \* t-shirt - 1  \* boxers - 1  White clothes:  \* briefs - 1  \* tanktop - 1 |
| 4  Red -> hat  Red -> dress,t-shirt,boxers  White -> briefs,tanktop  Blue -> gloves  White tanktop | Red clothes:  \* hat - 1  \* dress - 1  \* t-shirt - 1  \* boxers - 1  White clothes:  \* briefs - 1  \* tanktop - 1 (found!)  Blue clothes:  \* gloves - 1 |
| 5  Blue -> shoes  Blue -> shoes,shoes,shoes  Blue -> shoes,shoes  Blue -> shoes  Blue -> shoes,shoes  Red tanktop | Blue clothes:  \* shoes - 9 |

## \* Влогърите

Създайте програма, в която трябва да пазите информация за **влогъри** и техните **последователи**. На входа ще има **команди**. Те могат да бъдат в следния формат:

* "{Име на влогър}" **joined The V-Logger** – Добавя влогър.
  + Името на влогърът **ще съдържа само една команда**.
  + Ако името на **влогъра** съществува**, игнорирайте** командата**.**
* "{Влогър} followed {влогър}" – Първия влогър следва втория влогър.
  + Ако **някой** от влогърите не съществува, игнорирайте командата.
  + Влогър **не може** да следва **себе си**
  + Влогър **не може** да следва някой, **който вече го е последвал**
* **"Statistics" -** Когато получите тази командата, трябва да отпечатате статистика за влогърите

Всеки влогър има уникално име. Влогър може да **следва други влогъри** и да бъде **последван от влогъри, но** **не може да се следва** или **някой, който вече следва.** Трябва да отпечатате **броя на влогърите** в колекцията. След това трябва да отпечатате **най-известните влогъри** - тези с най-много последователи. Ако има повече от един влогъри с **еднакъв брой последователи**, отпечатайте първо човека, който следва **по-малко хора**. Влогърите трябва да бъдат отпечатани в **лексикографска ред**. Накрая отпечатайте останалите влогари подредени по **броя на последователите** в **низходящ ред**. След това по **броя на хората, които следват** във **възходящ ред**. Целият изход трябва да бъде в следния формат:

**"****The V-Logger has a total of {броя на влогърите} vloggers in its logs.**

**1.** {Най-известния влогър} : {последователи} followers, {последнивани} following

**\*** {последовател1}

\* {последовател2} …

**{Номер}.** {Влогър} : {последователи} followers, {последнивани} following

**{Номер}.** {Влогър} : {последователи} followers, {последнивани} following…"

### Вход

* Входа ще бъде във формата по-горе.

### Изход

* На **първият ред** ще отпечатате **броя на влогърите** във формата по-горе.
* На **вторият ред** отпечатайте **най-известния влогър**.
* На **следващите редове** отпечатайте **останалите влогъри**

### Бележки

* Няма да има ситуация, в която **два влогъра** да имат еднакъв брой **последователи** и **последвани**

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| EmilConrad joined The V-Logger  VenomTheDoctor joined The V-Logger  Saffrona joined The V-Logger  Saffrona followed EmilConrad  Saffrona followed VenomTheDoctor  EmilConrad followed VenomTheDoctor  VenomTheDoctor followed VenomTheDoctor  Saffrona followed EmilConrad  Statistics | The V-Logger has a total of 3 vloggers in its logs.  1. VenomTheDoctor : 2 followers, 0 following  \* EmilConrad  \* Saffrona  2. EmilConrad : 1 followers, 1 following  3. Saffrona : 0 followers, 2 following |

|  |  |
| --- | --- |
| JennaMarbles joined The V-Logger  JennaMarbles followed Zoella  AmazingPhil joined The V-Logger  JennaMarbles followed AmazingPhil  Zoella joined The V-Logger  JennaMarbles followed Zoella  Zoella followed AmazingPhil  Christy followed Zoella  Zoella followed Christy  JacksGap joined The V-Logger  JacksGap followed JennaMarbles  PewDiePie joined The V-Logger  Zoella joined The V-Logger  Statistics | The V-Logger has a total of 5 vloggers in its logs.  1. AmazingPhil : 2 followers, 0 following  \* JennaMarbles  \* Zoella  2. Zoella : 1 followers, 1 following  3. JennaMarbles : 1 followers, 2 following  4. PewDiePie : 0 followers, 0 following  5. JacksGap : 0 followers, 1 following |