# Упражнения: Комплекти и речници за напреднали

Задачи за [курса "C# Advanced" @ SoftUni](https://softuni.bg/trainings/4223/csharp-advanced-september-2023)   
Можете да проверите решенията си в [Judge](https://judge.softuni.org/Contests/1466/Sets-and-Dictionaries-Advanced-Exercise)

## Уникални потребителски имена

Създайте програма, която чете от конзолата поредица от **N потребителски имена** и съхранява колекция само от **уникалните** . На **първия ред ще** ви бъде дадено цяло число **N.** На следващия **Н** редове, ще получите **едно** потребителско име **на всеки** **линия** . Отпечатайте колекцията на конзолата по **реда** на **вмъкване** :

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 6  Джон  Джон  Джон  Петър  Джон  Момче1234 | Джон  Петър  Момче1234 |
| 10  Петър  Мария  Петър  Джордж  Сам  Мария  Сара  Петър  Сам  Джордж | Петър  Мария  Джордж  Сам  Сара |

## Набори от елементи

Създайте програма, която отпечатва **набор от елементи** . На първия ред ще получите две числа - **n** и **m** , които представляват дължините на две отделни групи. На следващия **n** + **м** редове, ще получите **n** числа, които са числата в **първия** набор, и **m** числа, които са във **втория** набор. Намерете всичко **уникално** **елементи** , които се появяват и в **двете,** и ги **отпечатва** в реда, в който се появяват в **първия** набор - **n** .

**Например:**

Набор с дължина n = 4: {1, **3** , **5** , 7}

Набор с дължина m = 3: { **3** , 4 , **5** }

Комплект, който съдържа всички **елементи** , които се повтарят и в **двете** **комплекти** -> { **3** , **5** }

### Примери

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

## Периодичната таблица

Създайте програма, която запазва всички **уникални** химически **елементи** . На първия ред ще ви бъде дадено число **n** - **броят** на входните **редове** , които ще получите. На следващите **n** реда ще получите **химикал** **съединения** , разделени с **един** **пространство** . Вашата задача е да отпечатате всички **уникални** във **възходящ ред** **поръчка** :

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4  Изпълнителен директор  Mo O Ce  Еее  мо | Ce Ee Mo O |
| 3  Ge Ch O Ne  Nb Mo Tc  О Не | Ch Ge Mo Nb Ne O Tc |

## Дори пъти

Създайте програма, която **отпечатва** число от колекция, което **изглежда четно** **брой** пъти **в** него. На първия ред ще ви бъде дадено **n** – **броят** на **целите числа ,** които ще получите. На следващите n реда ще получите **числата** . **Гарантирано** е , че **само** **един** от **тях изглежда** равен **брой** пъти. Вашата задача е да **намерите** това **число** и да го **отпечатате** накрая.

### Примери

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

## Преброяване на символи

**Създайте** програма, която чете някакъв **текст** от конзолата и **брои** срещанията на **всеки** знак в него. Отпечатайте резултатите по **азбучен ред** ( лексикографски ) ред.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| SoftUni скали | : 1 път/сек  S: 1 път/сек  U: 1 път/сек  c: 1 път/сек  f: 1 път/сек  i: 1 път/сек  k: 1 път/сек  n: 1 път/сек  o: 2 пъти/сек  r: 1 път/сек  s: 1 път/s  t: 1 път/сек |
| Знаете ли, че Math.Round закръгля до най-близкото четно цяло число? | : 9 пъти/сек  .: 1 път/сек  ?: 1 път/сек  D: 1 път/сек  M: 1 път/сек  R: 1 път/сек  a: 2 пъти/сек  d: 3 пъти/сек  e: 7 пъти/сек  g: 1 път/сек  ч: 2 пъти/сек  i: 2 пъти/сек  k: 1 път/сек  n: 6 пъти/сек  o: 5 пъти/сек  r: 3 пъти/сек  s: 2 пъти/s  t: 5 пъти/сек  u: 3 пъти/сек  v: 1 път/сек  w: 1 път/s  y: 1 път/сек |

## Гардероб

Създайте програма, която ви помага да решите какви **дрехи** да носите от **гардероба си** . Ще получите **дрехите** , които в момента са във вашия гардероб, сортирани по **цвят** в следния **формат** :

" {color} -> {item1},{item2},{item3}… "

Ако получите определен цвят, който вече **съществува** в гардероба ви, просто **добавете** дрехите към **него** **записи** . Можете също така да получавате **повтарящи се** **елементи** за определен **цвят** и трябва да поддържате техния **брой** .

Накрая ще получите **цвят** и дреха , която ще **търсите** в гардероба, разделени с интервал в следния формат **:**

**„{цвят} {дрехи}“**

Вашата задача е да отпечатате всички **елементи** и техния **брой** за **всеки** **цвят** в следния формат **:**

**"{цвят}** **дрехи** :

\* **{item1}** - **{count}**

\* **{item2}** - **{count}**

\* **{item3}** - **{count}**

…

\* **{itemN}** - **{count}"**

Ако намерите **артикула, който търсите** , трябва да отпечатате **"(намерен!)"** до него:

„\* **{itemN}** – **{count} (намерени!)“**

### Вход

* На **първия** **линия,** ще получите n - **броя на линиите** дрехи, които ще получите.
* На следващите **n** реда ще получите **дрехите** във **формата** **описано** по-горе.

### Изход

* Отпечатайте **дрехите** от вашия гардероб във **формат** **описано** по-горе.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4  Синьо -> рокля, дънки, шапка  Злато -> рокля, тениска, боксерки  Бял -> слип, потник  Син -> ръкавици  Синя рокля | Сини дрехи:  \* рокля - 1 (намерена!)  \* дънки - 1 бр  \* шапка - 1  \* ръкавици - 1 бр  Златни дрехи:  \* рокля - 1 бр  \* тениска - 1  \* боксерки - 1 бр  Бели дрехи:  \* слипове - 1 бр  \* потник - 1 |
| 4  Червено -> шапка  Червено -> рокля, тениска, боксерки  Бял -> слип, потник  Син -> ръкавици  Бял потник | Червени дрехи:  \* шапка - 1  \* рокля - 1 бр  \* тениска - 1  \* боксерки - 1 бр  Бели дрехи:  \* слипове - 1 бр  \* потник - 1 (намерен!)  Сини дрехи:  \* ръкавици - 1 бр |
| 5  Синьо -> обувки  Синьо -> обувки, обувки, обувки  Синьо -> обувки, обувки  Синьо -> обувки  Синьо -> обувки, обувки  Червен потник | Сини дрехи:  \* обувки - 9 бр |

## \*V-Logger

Създайте програма, която съхранява информацията за **влогърите** и техните **последователи** . Входът ще дойде като поредица от низове, където всеки низ ще представлява **валидна команда** . Командите ще бъдат представени в следния формат:

* " {vloggername} " **се присъедини към The V-Logger** – запазете влогъра в архивите си.
  + Имена на Vlogger **състоя се** **само от една дума** .
  + Ако **даденото** vloggernameвече **съществува, игнорирайте** товакоманда **.**
* „ {vloggername} последва {vloggername} “ – Първият влогър последва втория влогър.
  + Ако **някое** от **дадените имена на vloge** **не съществува** във вашата колекция, **игнорирайте** товакоманда **.**
  + Влогърът **не може да** следва **себе си.**
  + Влогърите **не могат** да следват някой, който **вече е последовател** **на.**
* **" Статистика " -** След като получите тази команда, трябва да отпечатате статистика за влогърите.

Всеки влогър има уникално име **на влогър** . **Влогърите** могат **да следват други влогъри** и влогърът **може да следва** **както много други влогъри** **както иска** , но **не може** да следва **себе си** или да следва някой, когото **вече е последовател** **от** . Трябва да отпечатате **общата сума** **брой** влогъри във вашата колекция **.** Тогава трябва да печатате най- **много** **известен** **vlogger** – този с най-много последователи, с **неговия** **последователи.** Ако повече от един влогъри имат **същото** **брой** последователи **,** отпечатайтеедин **следва по-малко** хора, а неговите **последователи** трябва да бъдат отпечатани в **лексикографски ред (** в случай че влогърът няма **последователи** , отпечатайте само първия ред, който е описан **по-долу** ). И накрая, отпечатайте **останалите** влогъри , подредени по **броя** на последователите в **низходящ** ред, след това по броя на влогърите, които той следва във **възходящ ред** **поръчка.** Целият **резултат трябва да бъде** в следния формат:

**"** **V-Logger има общо {registered vloggers} влогъри в своите регистрационни файлове.**

**1.** {mostFamousVlogger}: {followers} последователи, {followings} следващи

**\*** {follower1}

\* {follower2} …

**{Не}.** {vlogger} : {followers} последователи, {followings} следващи

**{Не}.** {vlogger} : {followers} последователи, {followings} следващи…"

### Вход

* Въведеното ще дойде във формата, описан по-горе.

### Изход

* На първия ред отпечатайте **общия брой влогъри** във формата, описан по-горе.
* На втория ред отпечатайте най- **много** **известен** влогър във формата, описан по-горе.
* На **следващите** редове отпечатайте всички **останали** влогъри във формата, описан по-горе.

### Ограничения

* **Няма** да има **невалиден** вход.
* Няма да има ситуация, в която **две** **влогърите** имат **равен** брой **последователи** и **равен** брой **следвания.**
* Разрешено време/памет: **100ms/ 16MB** .

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| EmilConrad се присъедини към The V-Logger  VenomTheDoctor се присъедини към The V-Logger  Saffrona се присъедини към The V-Logger  Шафрона последва Емил Конрад  Saffronа последва VenomTheDoctor  EmilConrad последва VenomTheDoctor  VenomTheDoctor последва VenomTheDoctor  Шафрона последва Емил Конрад  Статистика | V-Logger има общо 3 влогъри в своите регистрационни файлове.  1. VenomTheDoctor : 2 последователи, 0 следващи  \* Емил Конрад  \* Шафран  2. EmilConrad : 1 последователи, 1 следване  3. Saffrona: 0 последователи, 2 следващи |
| JennaMarbles се присъедини към The V-Logger  ДженаМарбълс последва Зоела  AmazingPhil се присъедини към The V-Logger  JennaMarbles последва AmazingPhil  Zoella се присъедини към The V-Logger  ДженаМарбълс последва Зоела  Zoella последва AmazingPhil  Кристи последва Зоела  Зоела последва Кристи  JacksGap се присъедини към The V-Logger  JacksGap последва JennaMarbles  PewDiePie се присъедини към The V-Logger  Zoella се присъедини към The V-Logger  Статистика | V-Logger има общо 5 влогъри в своите регистрационни файлове.  1. AmazingPhil: 2 последователи, 0 следващи  \* Джена Марбълс  \* Зоела  2. Zoella: 1 последователи, 1 следване  3. JennaMarbles: 1 последователи, 2 следващи  4. PewDiePie: 0 последователи, 0 следващи  5. JacksGap: 0 последователи, 1 следване |

## \*Класиране

Създайте програма, която **класира** кандидат-стажантите, в зависимост от **точките** от **интервюто** **задачи** и техния **изпит** **резултати** в СофтУни . Ще получите няколко реда за **въвеждане** във формат **„{конкурс}:{парола за конкурс}“** докато не получите **"край на конкурсите "** . Запазете тези данни, защото **ще ви трябват по-късно** . След това ще получите друг тип въведени данни във формат **"{contest}=>{password}=>{username}=>{points}"** докато не получите **"край на подаванията "** . Ето какво трябва да направите:

* Проверете дали **състезанието е валидно ( ако сте го получили при първия тип въвеждане ).**
* Проверете дали **паролата е правилна за дадения конкурс.**
* Запазете потребителя със **състезанието** , в което участва **( потребителят може да участва в много състезания )** и точките, които потребителят има в **дадения** **състезание** \_ Ако получите **същия конкурс** исъщия **потребител, актуализирайте** точките **само ако новите** са **повече от** по **-старите.**

Накрая трябва да отпечатате информацията за потребителя с най- **много точки** във формата:

**„Най-добрият кандидат е {user} с общо {total points} точки.“** . След това отпечатайте **всички ученици, подредени** потехните **имена** . За **всеки** потребител,отпечатайте **всеки конкурс** сточките в **низходящ** ред в следния формат **:**

**"{user1 name}**

**# {конкурс1} -> {точки}**

**# {конкурс2} -> {точки}**

**{user2 име}**

**…"**

### Вход

* Ще получавате низове във форматите, описани по-горе, докато не бъдат дадени подходящите команди, също описани по-горе.

### Изход

* На **първия** ред отпечатайте най-добрия потребител във формата, **описан** по-горе.
* На **следващите** редове отпечатайте всички ученици, подредени, както е споменато по-горе във формат.

### Ограничения

* **Няма** да има случай с две **равни** **състезания** .
* Низовете могат да съдържат всеки ASCII знак с изключение на **(:, =, >) .**
* Числата ще бъдат в диапазона **[ 0…10000]** .
* Вторият вход винаги **е валиден** .
* Няма да има случай с **2** или **повече** потребители с **едно и също** **обща сума** **точки** .

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Първа част Интервю: успех  Js Основи: JSFundPass  C# Основи: fundPass  Алгоритми: забавление  край на състезанията  C# Fundamentals=>fundPass=>Tanya=>350  Алгоритми=>забавление=>Таня=>380  Първа част Интервю=>успех=>Никола=>120  Изпит по основи на Java=>JSFundPass=>Parker=>400  Първа част Интервю=>успех=>Таня=>220  OOP Advanced=>password123=>BaiIvan=>231  C# Fundamentals=>fundPass=>Таня=>250  C# Fundamentals=>fundPass=>Nikola=>200  Js Fundamentals=>JSFundPass=>Таня=>400  край на подаванията | Най-добър кандидат е Таня с общо 1350 точки.  класиране:  Никола  # C# Основи -> 200  # Част първа Интервю -> 120  Таня  # Js Основи -> 400  # Алгоритми -> 380  # C# Основи -> 350  # Част първа Интервю -> 220 |
| Java Advanced: funpass  Втора част Интервю: успех  Математическа концепция: asdasd  Java Web Basics: forrF  край на състезанията  Math Concept=>ispass=>Monika=>290  Java Advanced=>funpass=>Simon=>400  Втора част Интервю=>успех=>Драго=>120  Java Advanced=>funpass=>Petyr=>90  Java Web Basics=>forrF=>Simon=>280  Втора част Интервю=>успех=>Petyr=>0  Математическа концепция=>asdasd=>Drago=>250  Втора част Интервю=>успех=>Саймън=>200  край на подаванията | Най-добър кандидат е Саймън с общо 880 точки.  класиране:  Драго  # Математическа концепция -> 250  # Част втора Интервю -> 120  Петър  # Java Advanced -> 90  # Част втора интервю -> 0  Саймън  # Java Advanced -> 400  # Java Web Basics -> 280  # Част втора Интервю -> 200 |

## \*SoftUni Резултати от изпита

Съдийската статистика на последния изпит Programing Fundamentals не работеше правилно, така че имате задачата да вземете всички изпратени документи и да ги анализирате правилно. Трябва да съберете всички изпратени документи и да отпечатате крайните резултати и статистика за всеки език, на който участниците са изпратили своите решения.

Ще получавате редове в следния формат: „{потребителско име}-{език}-{точки}“ , **докато не получите** „изпитът е завършен“ .Трябва да съхранявате всяко потребителско име и неговите заявки и точки.   
Можете да получите **команда за банване** на потребител за измама в следния формат: "{username}-banned" . В такъв случай трябва да **премахнете** потребителя от състезанието, но **да запазите неговите предложения в общия брой на изпращанията за всеки език** .

След получаване на „изпитът е завършен“, отпечатайте всеки от участниците, подредени в низходящ ред по техните максимални точки, след това по потребителско име, в следния формат:

" Резултати: "

" {потребителско име} | {точки} "

…

След това отпечатайте всеки език, използван в изпита, подреден в низходящ ред по общия брой подавания и след това по име на език в следния формат:

" Изпращания: "

" {език} – {submissionsCount} "

…

### Вход / Ограничения

Докато не получите „ **изпитът е завършен** “, ще получавате заявки за участници в следния формат: „{потребителско име}-{език}-{точки}“ .

Можете да получите команда за забрана -> "{username}-banned"

Точките на участника винаги ще бъдат **валидно цяло число в диапазона [0-100];**

### Изход

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

" Резултати: "

" {потребителско име} | {точки} "

…

* След това отпечатайте всеки език, подреден в низходящ ред по общия брой подавания и след това по име на език, в следния формат:

" Изпращания: "

" {език} – {submissionsCount} "

…

* Разрешено работно **време** / **памет** : **100ms** / **16MB** .

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментирайте** |
| Петър-Ява-84  Георги-C#-70  Георги-C#-84  Сам-C#-94  изпитът приключи | Резултати:  Сам | 94  Джордж | 84  Петър | 84  Изпращания:  C# - 3  Java - 1 | Ние подреждаме участниците по максимален брой точки и след това по име, като отпечатваме само потребителското име и максималния брой точки.  След това отпечатваме всеки език заедно с броя на изпращанията, подредени в низходящ ред по брой изпращания и след това по име на език. |
| Петър-Ява-91  Георги-C#-84  Sam-JavaScript-90  Сам-C#-50  Сам забранен  изпитът приключи | Резултати:  Петър | 91  Джордж | 84  Изпращания:  C# - 2  Java - 1  JavaScript - 1 | Сам е забранен, така че е отстранен от състезанието, но неговите предложения все още се запазват в броя на изпратените езици.  Така че въпреки че има само 2 участници в резултатите, има общо 4 заявки. |

## \*ForceBook

Потребителите на сила се борят да запомнят от коя страна са различните потребители на сила, защото ги сменят твърде често. Така че имате задачата да създадете уеб приложение за управление на техните профили. Трябва да съхранявате информация за всеки **уникален forceUser** , регистриран в приложението.

Ще получите **няколко входни реда** в един от следните формати:

{forceSide} | {forceUser}

{forceUser} -> {forceSide}

ForceUser и forceSide са низове, съдържащи произволен знак.

Ако получите forceSide | forceUser , трябва да **проверите дали такъв forceUser вече съществува** и **ако не** , **добавете** го/я към съответната страна.

Ако получите forceUser -> forceSide , трябва да проверите дали вече има такъв **forceUser** и ако е така, **сменете неговата/нейната страна** . Ако няма такъв **forceUser** , добавете го/я към съответната forceSide, третирайки командата **като новорегистриран forceUser.**   
След това трябва да отпечатате на конзолата: "{forceUser} се присъединява към {forceSide} страна!" .

Трябва да прекратите програмата си, когато получите командата "Lumpawaroo" . В този момент трябва да отпечатате всяка силна страна, **подредена в низходящ ред по броя на forceUsers, след това подредена по име** . За всяка страна отпечатайте forceUsers **,** подредени **по име** .

В случай че няма **forceUsers отстрани** , **не трябва да отпечатвате** страничната информация.

### Вход / Ограничения

* Входът идва под формата на команди в един от форматите, посочени по-горе.
* Въвеждането приключва, когато получите командата "Lumpawaroo" .

### Изход

* Като изход за всеки forceSide, **подреден в низходящ ред по брой на forceUsers** , **след това по име** , трябва да отпечатате всички forceUsers, **подредени по име по азбучен ред** .
* Изходният формат е:

" Страна: {forceSide}, Членове: {forceUsers.Count} "

" ! {forceUser} "

" ! {forceUser} "

" ! {forceUser} "

* В случай че **НЯМА** forceUsers , не отпечатвайте тази страна.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| Светлина | Джордж  Тъмно | Петър  Лумпавару | Страна: Тъмно, Членове: 1  ! Петър  Страна: Светли, Членове: 1  ! Джордж | Регистрираме Георги в Светлината страна и Пешо в Мрака страна . След получаване на " Lumpawaroo " ние отпечатваме двете страни, подредени по membersCount и след това по име. |
| Запалка | Кралски  По-тъмен | DCay  Джон Джонис -> Запалка  DCay -> Запалка  Лумпавару | Джон Джонис се присъединява към по-леката страна!  DCay се присъединява към по-леката страна!  Страна: Запалка, Членове: 3  ! DCay  ! Джон Джонис  ! Кралски | Въпреки че Джон Johnys няма профил, ние го **регистрираме** и го добавяме към Lighter страна.  Премахваме **DCay** от Darker **\_** страна и го добавете към Lighter страна .  Ние печатаме само Lighter страна , защото Darker страна **няма членове.** |