# Упражнения: Множества и речници

Тествайте решението в Judge: <https://judge.softuni.org/Contests/4161/05-Sets-and-Dictionaries-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  Mo O Ce  Ce O  Fe  Mo | Ce Fe Mo O | 3  Ge Ch O Ne  Nb Mo Tc  O Ne | Ch Ge Mo Nb Ne O Tc |

### Насоки

1. Прочетете броя на **химичните съединения**.
2. Създайте **сортиран сет** от **низове**, за да съхранявате **уникалните имена** като ги запазите сортирани.
3. Създайте for-цикъл, за да прочетете и след това да добавите **всички химични съединения**.
4. Накрая отпечатайте **всички елементи**.

## 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 |

### Насоки

1. Прочетете **броя на числата**.
2. Създайте **речник**, чрез който ще съхранявате **броя на срещаните числа**.
3. Създайте for**-цикъл**, за да **прочетете** и **добавите** **числата** в **речника** като при всяко срещане на число **увеличете** срещането му с **1**.
4. Накрая създайте foreach-цикъл, за да обходите **всички елементи** в **речника** и да **отпечатате** **числото**, което се среща четен брой пъти.

## 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 чрез събиране на материали. Нямате претенции и затова ще вземете продукта 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: 255  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 |

### Насоки

1. Създайте **речник**, в който ще пазите **материалите** и техния **брой**.
2. Създайте още един **речник**, в който ще пазите **отпадъците** и техния **брой**.
3. Създайте **низ**, чрез който ще прочитате входните данни.
4. След това в него създайте while**-цикъл**, който ще итерира докато **не свърши** входа.
5. В него превърнете **входа** в **масив**.
6. След това създайте още един for**-цикъл**, който ще итерира докато неговия брояч е **различен от дължината** на масива.
7. След всяко итериране на for**-цикъла**, увеличете неговия брояч с **2**.
8. В него добавете **материала** или **боклука** в **речника**.
9. Накрая **отпечатайте продуктите**, които сте получили с материалите от входа.
10. Останалите материали ги отпечатайте, **сортирани** по **количество във възходящ ред**

## 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:
  + Регистрaция: "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 |