# Контролна работа

Тествайте решението в Judge: <https://judge.softuni.org/Contests/4304/07-Mid-Exam>.

## Пътешествие

*Вие сте пътешественик по света и следващата ви цел е да направите околосветска обиколка. За да направите това, първо трябва да планирате всичко. За да започнете, трябва да планирате къде ще бъдат вашите почивки.*

На **първия ред** ще получите низ, съдържащ всичките **спирки**. Докато не получите командата **"Travel"**, ще получавате команди за **манипулиране** на първоначалния низ. **Командите** са:

* **"Add Stop:{индекс}:{низ}"**:
  + Ако индексът е **валиден**,командата **добавя низа** на **индекса**
* **"Remove Stop:{начален индекс}:{краен индекс}"**:
  + Ако индексите са **валидни**, командата **премахва** елементи на низ от стартовия индекс до крайния индекс (**включително**)
* **"Switch:{стар низ}:{начален низ}"**:
  + Ако **старият низ** се съдържа в началния, **заменете** го **навсякъде** **с новия низ**

***Бележка: След всяко добавяне, отпечатайте низа в сегашното му състояние!***

След като получите командата **"Travel"**, отпечатайте:**"Ready for world tour! Planned stops: {низ}"**

### Вход

* **Индексът е валиден**, когато е межу **пърия** и **последен** символ на низа (включително).

### Изход

* Изходът трябва да бъде правилно изведен, както е описан по-горе

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Hawai::Cyprys-Greece  Add Stop:7:Rome  Remove Stop:11:16  Switch:Hawai:Bulgaria  Travel | Hawai::RomeCyprys-Greece  Hawai::Rome-Greece  Bulgaria::Rome-Greece  Ready for world tour! Planned stops: Bulgaria::Rome-Greece |
| Albania:Bulgaria:Cyprus:Deuchland  Add Stop:3:Nigeria  Remove Stop:4:8  Switch:Albania: Azərbaycan  Travel | AlbNigeriaania:Bulgaria:Cyprus:Deuchland  AlbNaania:Bulgaria:Cyprus:Deuchland  AlbNaania:Bulgaria:Cyprus:Deuchland  Ready for world tour! Planned stops: AlbNaania:Bulgaria:Cyprus:Deuchland |

## Унищожаване на чудовища



*Трябва да симулирате битка между войник и група от ужасяващи чудовища. Войникът има поразителни удари, а чудовищата имат различно ниво на броня. Вашата задача е да нашите програма, която взема две редици от числа като вход от конзолата и да симулирате битка.*

Ще получите **две редици от числа**.

**Първата редица** представлява бронята на чудовищата. **Всяко число** представлява **бронята на едно чудовище**.

Втората редица представлява поразителните действия на войника. **Всяко число представлява силата** на удара, **нанесен от войника**.

### Правила за битка

* Чудовището в **началото на редицата** ще бъде **първият**, който ще се **сбие с войника**. Взимаме първата броня, **последната сила на удара** и започваме да **сравняваме стойностите**.
* Ако силата на удара е по-голяма от или равна на бронята на чудовището, **убиваме чудовището** и **премахваме неговата броня** от редицата. Ударът на войника се **намаля** със стойността на бронята на чудовището. Оставащата сила се **добавя** към **следващата сила** от редицата (ако има) или се зачита за **последен и единствен елемент**. **Нулевите** стойности **не трябва да се връщат** обратно към редицата.
* Ако силата на удара е **по-малка** от бронята на чудовището, ударът се **изпълнява** и се **премахва** от редицата, но бронята на чудовището се **намалява със силата на удара**. Чудовището се премества в **края на редицата**.
* Битката приключва, **когато една от редиците стане празна**.

### Вашата задача

Напишете конзолно приложение, което симулира битката по-горе. Имплементирайте логиката, използвайки **правилните структури от данни** за управляване на ударите на войника и на броните на чудовищата. Програмата трябва да покаже подходящия резултат от битката въз основа на правилата.

### Вход

* На първия ред ще получите **стойностите на броните на чудовищата** – цели числа, разделни със запетая.
* На втория ред ще получите **стойностите на ударите на войника** – цели числа, разделни със запетая.

### Изход

* Ако **всички чудовища са убити**, програмата трябва да отпечата на конзолата:
  + **"All monsters have been killed!"**
* Ако **войникът** **остане** **без удари**, програмата трябва да отпечата на конзолата:
  + **"The soldier has been defeated."**
* Накрая програмата отпечатва **колко чудовища е убил войникът на нов ред**:
  + **"Total monsters killed: {броя убити чудовища}"**

### Бележки

* Всички числа ще бъдат в обхвата **[1, 100]**
* Няма да има **отрицателни числа** във входа

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 20,15,10  5,15,10,25 | All monsters have been killed!  Total monsters killed: 3 |
| **Обяснения** | |
| 1. Първият удар на войника (последния в редицата) (25) е по-голяма от първата броня на чудовището (20). **Първото чудовище** е **убито**. Неговата броня е премахната от редицата. Силата на войникът се намалява с бронята на чудовището и получава стойност (5). Добавяме го към следващия елемент в редицата (10) 2. Вторият удар на войника (15 {10 + 5}) е равен на втората броня на чудовището(15). Второто чудовище е убито и се **премахва неговата броня** от редицата. Удара на войника се намалява с бронята на чудовището и става на (0). Не е необходимо да го добавяме към следващия елемент. Накрая **премахваме удара**. 3. Третия удар на войника (15) е по-голям от третата броня на чудовището (10). Следователно третото чудовище умира. Неговата броня се премахва от редицата. Заради това редицата става празна и отпечатваме изхода на конзолата: **"**All monsters have been killed.   The soldier wins the battle, having killed 3 monsters.**"** | |
| **Вход** | **Изход** |
| 30,25,40,35  15,20,10,30 | The soldier has been defeated.  Total monsters killed: 1 |
| **Обяснения** | |
| 1. Първия удар на войника (30) е равен на бронята на първото чудовище (30). **Чудовището е убито** и **бронята** му се премахва от редицата, а удара на войника е равен на нула, затова не я добавяме към следващия удар и премахваме от редицата. 2. Втория удар на войника (10) е по-малък от бронята на второто чудовище (25) и оцелява с оставаща броня 15, затова **отива в края на редицата**. 3. Третата атака на войника (20) е по-малка от бронята на третото чудовище (40). Чудовището оцелява с оставаща броня 20. 4. Четвъртата атака на войника (15) е по-малка от бронята на четвъртото чудовище (35). Чудовището оцелява с оставаща броня 20. 5. Войникът няма повече удара и има още 3 чудовища. | |

## 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 команда "**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}**"

* Курсът и името 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 студент}"**

### 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 |