# Практически задачи: задачи върху масиви

## Обработка на масив

Вие ще получите масив от низове и трябва да изпълните командите под тях. Вие можете да получите три команди:

* Reverse – обръща реда в масива
* Distinct – изтрива всички неуникални (повтарящи се) елементи на масива
* Replace {index} {string} – замества елемента на дадената позиция index с низ string, който ви е даден

Вход

* На първи ред, получавате масив от символни низове string array
* На втори ред ще получите броя на редове, които следват
* На следващите n реда ще получите команди

Изход

Изведете масива в следния формат:

{1st element}, {2nd element}, {3rd element} … {nth element}

Ограничения

* За разделител да се използва един инервал
* N ще е цяло число в интервала [1…100]

Примери

|  |  |
| --- | --- |
| Вход | Изход |
| one one one two three four five  3  Distinct  Reverse  Replace 2 Hello | five, four, Hello, two, one |

## Безопасна обработка на масив

Сега ние трябва да направим нашата програма, по-безопасна и по-лесна. Направете програмата да отпечата "Невалиден вход!", ако ние се опитваме да замените елемент с несъществуващ индекс или се изписва „невалидна команда”, ако командата не е валидна. Също така да работи до командата "Край".

Вход

* На първи ред ще получите входния масив от низове
* На следващите редове ще получавате команди, докато не получите команда “END” за край

Изход

На края изведете масива в следния формат:

{1st element}, {2nd element}, {3rd element} … {nth element}

Ограничения

* Само един интервал да се ползва за разделител.
* n ще е цяло число integer в интервала [1…100]

Примери

|  |  |
| --- | --- |
| Вход | Изход |
| one one one two three four five  Distinct  Reverse  Replace 7 Hello  Replace -5 Hello  Replace 0 Hello  END | Invalid input!  Invalid input!  Hello, four, three, two, one |
| Вход | Изход |
| Alpha Bravo Charlie Delta Echo Foxtrot  Distinct  Reverse  Replace 0 Charlie  Reverse  Replace 1 Charlie  Distinct  Replace 4 Charlie  END | Invalid input!  Invalid input!  Alpha, Charlie, Delta, Echo |

## Множество от сумите на последните k числа

Въведете две числа n и k. Създайте и изведете следното множество от n елемента:

* Първият елемент е : 1
* Всички други елементи са = сбор от предишните k елемента (ако няма k елемента преди текущия, то да се изведе сбора на всички до момента)
* Пример: n = 9, k = 5 🡪 120 = 4 + 8 + 16 + 31 + 61

Примери

|  |  |
| --- | --- |
| Вход | Изход |
| 6  3 | 1 1 2 4 7 13 |
| 8  2 | 1 1 2 3 5 8 13 21 |
| 9  5 | 1 1 2 4 8 16 31 61 120 |

Упътване

* Използвайте масив от цели числа, в който да пазите числата.
* Инициализирайте първия елемент seq[0] = 1
* Използвайте два вложени цикъла:
  + Преминете през всички елементи в цикъл за i = 1 … n
  + Съберете елементите i-k … i-1: seq[i] = sum(seq[i-k … i-1])

## Извличане на средните 1, 2 или 3 елемента

Напишете метод за извличане на средните 1,2 или 3 елемента от масив от цели числа и изведете резултата на конзолата

* n = 1 -> 1 елемент
* за четно n -> 2 елемента
* за нечетно n -> 3 елемента

Създайте програма, която чете масив от цели числа (разделени с интервал) и отпечатва средните елементи във формата показан в примерите

Примери

|  |  |
| --- | --- |
| Вход | Изход |
| 5 | { 5 } |
| 2 3 8 1 7 4 | { 8, 1 } |
| 1 2 3 4 5 6 7 | { 3, 4, 5 } |
| 10 20 30 40 50 60 70 80 | { 40, 50 } |

Упътване

* Напишете различна логика за всеки от 3 те случая (n = 1, четно n, нечетно n)
* n = 1 🡪 взима първия елемент
* нечетно n 🡪 взема елементите n/2-1, n/2, n/2+1
* четно n 🡪 взема елементте n/2-1 и n/2

## Склад

Ще ви бъдат дадени три масиви на различни редове. Първият ще съдържа низове, които ще представляват имената на продуктите. Вторият ще съдържа големи цели числа longs и ще представляват количествата на продуктите. Третият ще съдържа дробни числа, които са цените на продуктите. След което ще бъдат дадени имена на продукти на нови редове, докато получите командата " Done". За всяко дадено име на продукт изведете :

{име на продукта} разходи: {цена}; Налично количество: {количеството}

Имената, цените и количествата на продуктите са с едни и същи индекси в 3 масива.

Вход

На трети ред вие ще получите масив с десетични числа, които представляват цените на продуктите.

Ограничения

Трите масива винаги ще имат една и съща дължина. Вие винаги ще получавате съществуващите продукти.

* На първия ред, вие ще получите масив от символни изове, които представляват имената на продуктите.
* На втори ред вие ще получите масив с дълги цели числа, които представляват количествата на продуктите.
* Третият ще съдържа дробни числа, които са цените на продуктите.

Примери

|  |  |
| --- | --- |
| Вход | Изход |
| Bread Juice Fruits Lemons  10 50 20 30  2.34 1.23 3.42 1.50  Bread  Juice  done | Bread costs: 2.34; Available quantity: 10  Juice costs: 1.23; Available quantity: 50 |
| Oranges Apples Nuts  1500 5000000 2000000000  2.3412 1.23 3.4250  Nuts  done | Nuts costs: 3.4250; Available quantity: 2000000000 |

Упътване

* В C#, вие можете да намерите индекса на елемент с Array.IndexOf(array, element)
* В Java, най-лесният начин да намерите индекса на елемент (без използване на външни библиотеки) ще е да проверите целия масив

## \* Склад обновена версия

За тази задача можете да използвате вашето решение от задачата за склада. Отново ще получите 3 масиви – един с низове, с цели числа longs и с дробни числа с десетични знаци. Отново цената и количеството съответства на името, което се намира на същия индекс като име. Този път само масива, съдържащ имената и масива, съдържащ цените ще имат същата дължина. Ако в масива с количествата няма индекс, който отговаря на името, трябва да се приеме количество 0. Освен това продуктите, които получавате след масиви ще съдържа не само низ за името, но и дълга, който е, количеството, което трябва да се поръча. Ако имате достатъчно количество, да се изчисли общата цена чрез умножаване на поръчаните количества по цената и да я отпечатате в следния формат:

{име на продукта} x {поръчано количество} струва {обща цена на поръчката}

Форматирайте цената до втория знак след десетичната запеая. Не забравяйте да намалите количеството на продукта. Ако нямате необходимото количество изведете:

We do not have enough {product name}

Вход

* На първи ред ще получите масив от низове, който съдържа имената на продуктите
* На втори ред ще получите масив от цели числа longs, който съдържа количествата на продуктите
* На трети ред ще получите масив от дробни числа , който съдържа цените на продуктите

Ограничения

* Масивите с имената и цените да са с еднаква дължина.
* Винаги се въвеждат съществуващи продукти

Примери

|  |  |
| --- | --- |
| Вход | Изход |
| Bread Juice Fruits Lemons Beer  10 50 20 30  2.34 1.23 3.42 1.50 3.00  Bread 10  Juice 5  Beer 20  done | Bread x 10 costs 23.40  Juice x 5 costs 6.15  We do not have enough Beer |
| Tomatoes Onions Lemons  10000 2000  5.40 3.20 2.20  Tomatoes 5000  Tomatoes 5000  Tomatoes 1  done | Tomatoes x 5000 costs 27000.00  Tomatoes x 5000 costs 27000.00  We do not have enough Tomatoes |