# Допълнителни упражнения: Списъци

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/3171/Additional-Exercises>.

## Еднакви числа

Напишете програма, която да събира всички съседни числа с еднаква сума в списъкът отляво надясно.

* След като **сумирате** **двете числа** , **резултатът** може да бъде равен на **съдеда** на второто число и трябва да се **сумира също** (както е показоно по-долу).
* Винаги сумирайте **най-левите два равни съседа** (ако има няколко равни двойки съседи)

Програмата свършва, когато няма повече съседни числа с еднакви стойности. Накрая отпечатайте крайният резултат.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3 3 6 1 | 12 1 | **3 3** 6 1 à **6 6** 1 à 12 1 |
| 8 2 2 4 8 16 | 16 8 16 | 8 **2 2** 4 8 16 à 8 **4 4** 8 16 à **8 8** 8 16 à 16 8 16 |
| 5 4 2 1 1 4 | 5 8 4 | 5 4 2 **1 1** 4 à 5 4 **2 2** 4 à 5 **4 4** 4 à 5 8 4 |

### Насоки

Първо четем списък от числа.

Text

Description automatically generated with medium confidence

Минаваме през всички елементи. Проверяваме дали числото със сегашният индекс и следващото са еднакви. Ако е така, нулираме for-цикъла и започваме отначало. В противен случей не правим нищо.

Text

Description automatically generated

Накрая отпечатваме всички числа **разделени с интервал**.



## Манипулация на списък

Напишете програма, която чете списък от числа. Докато не получите **"end",** ще получавате следните възможни **команди**:

**Add {число}:** Добавяне на число в края на списъка.  
**Remove {число}:** Премахване на число от списъка.  
**RemoveAt {индекс}:** Премахване на число с определен индекс.  
**Insert {число} {индекс}:** Вмъкване на число на определен индекс.

**Бележка: Не всичко ще бъде валидно!**

Когато получите **"end",** принтирате **финалния** списъка (**числата трябва да бъдат разделени с интервал**).

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4 19 2 53 6 43  Add 3  Remove 2  RemoveAt 1  Insert 8 3  end | 4 53 6 8 43 3 |

### Насоки

Първо прочитаме списъка от конзолата.

Text

Description automatically generated

* **Разделяме** низа получен на първия ред от конзолата. След това **минаваме през всеки елемент** и го парсваме към **число**.
* Това връща **IEnumarable<int>** (**колекция** от числа) и го запазваме под формата на списък.

След входа използваме while-цикъл и switch case за различните команди.

Graphical user interface, text, application

Description automatically generated

* Спираме цикъла, когато реда е “end”, в противен случай **разделяме** входа на низвое в **tokens**.

A picture containing table

Description automatically generated

Имплементираме всички **команди**.

Text

Description automatically generated

* За всички команди с **изключение** на **"Insert"** , **tokens[1]** е винаги **число/индекс**.
* За **"Insert"** командата ние получаваме **число и индекс** (**tokens[1], tokens[2]**).

Накрая отпечатваме числата, разделени с интервал.



## Манипулация на списъци (Advanced)

В тази задача ще трябва да имплентираме по-сложни команди на списъка(**разширяване на предходната задача**). Отново ще четете списък и команди, докато не получите **"end":**

**Contains {число}** – Проверявате дали число съществува в списъка - Ако е така принтирайте **"Yes**", в противен случай **"No such number ".**

**PrintEven** – принтирайте **всички четени числа разделени с интервал.**

**PrintOdd** – принтирайте **всички нечетени числа разделени с интервал.**  
**GetSum** – принтирайте **сумата на всички числа**.  
**Filter {condition} {number}** – отпечатайте всички числа, които отговарят на **даденото условие**. Условията могат да бъдат **'<', '>', ">=", "<="**.

След командата **"end"** принтирайте списъка, ако има някакви промени от оригиналния списък. **Промените** се правят само от командите от **предходната задача**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 2 13 43 876 342 23 543  Contains 100  Contains 543  PrintEven  PrintOdd  GetSum  Filter >= 43  Filter < 100  end | No such number  Yes  2 876 342  13 43 23 543  1842  43 876 342 543  2 13 43 23 |

## **Съобщения**

Ще ви бъде даден **списък от числа** и **низ**. За всеки елемент трябва да **съберете неговите цифри**, за да получите определен индекс от **низа**. Ако индексът е **по-голям от дължината**, започнете да броите от самото начало (така винаги ще има валиден индекс). След като **вземете елемента**, го премахнете от низа. Така за следващият индекс, низа ще бъде с един символ по-малко.

### **Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 9992 562 8933  This is some message for you | hey |

## **Състезание**

Напишете програма, която ще пресмята **коя кола ще победи в състезанието за коли**. Ще получите **списък от числа**. Всеки елемент представлява **необходимото време за всяка стъпка** (индекс). Ще има **две коли**. **Едната започва от лявата част**, а другото  **от дясната част**. Средата на масива ще бъде **финалната линия**. Броя на елементите винаги ще бъде **четен**. Пресметнете **необходимото време** на всеки състезател да премине **финалната лента**. Отпечатайте **времето на победителя**. Ако имате **0** в списъка, трябва да намалите текущото време **с 20%**. Отпечатайте резултата в следния формат : **"The winner is {left/right} with total time: {общо време}".**

### **Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 29 13 9 0 13 0 21 0 14 82 12 | The winner is left with total time: 53.8 |
| **Коментари** | |
| Времето на левият състезател е (29 + 13 + 9) \* 0.8 (заради нулата) + 13 = 53.8  Времето на десният състезател е (82 + 12 + 14) \* 0.8 + 21 = 107.4  Отпечатваме левия състезател, защото има по-малко време | |

## **Скрити съобщения**

Напишете програма, която чете **низ** и **открива** скритите съобщения. Трябва да приложете следният алгоритъм:

Нека нашият низ е “**skipTest\_String044170**”.

Запазваме цифрите от низа. След това ще премахнем всички числа от низа. В тази операция трябва да имате два списъка: списък с числа и списък без числа:

* Списък с числа: **[0, 4, 4, 1, 7, 0]**
* Списък без числа: **[s, k, i, p, T, e, s, t, \_, S, t, r, i, n, g]**

След това разделете списъка с числа на взевти (на четни позиции) и пропуснати (на нечетни позиции):

* Списък от пропуснатите: **[0, 4, 7]**
* Списък от взетите: **[4, 1, 0]**

След това **минаваме** през двата списъка. **Пропускаме** символи{**индекс от списъка на пропуснатите числа**} от списъка без числа и добавяме символи {**индекс от списъка на взетите числа**} в крайният низ. Процедурата трябва да изглежада последният начин:

1. Взимаме **0** символа è Вземаме: "", пропускаме **4** символа (общо **0**) è   
   Пропуснато: "**skipTest\_String**"è Резултат: ""
2. Взимаме **4** символаè Вземаме: "**Test"**, пропускаме **1** символ (общо **4**) è Пропуснато: "**skip**" è Резултат: "**Test**"
3. Взимаме **7** символаè Взимаме: "**String**", пропускаме **0** символа (общо **9**)è Пропуснато: "" è Резултат: "**TestString**"

Накрая отпечатваме крайният низ на конзолата.

### **Вход**

* На първия ред получавате **криптираното** съобщение като **низ**

### **Изход**

* На първия ред принтираме **декриптираното** съобщение като **низ**

### **Бележки**

* Броя на числата **винаги** ще бъде валиден.
* Криптираното съобщение винаги ще бъде от ASCII символи.

### **Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| T2exs15ti23ng1\_3cT1h3e0\_Roppe | TestingTheRope |
| O{1ne1T2021wf312o13Th111xreve!!@! | OneTwoThree!!! |
| this forbidden mess of an age rating 0127504740 | hidden message |

## **\*Миксиран списък**

Напишете програма, **която миксира два списъка**. Ще получите вход от **два реда**, като всеки представлява **списък от числа**. Правилата за смесване са:

* Започнете от **началото на първия списък** и от **края на втория списък.**
* **Добавете** елементи от **първия** и от **втория**.
* Накрая винаги ще има списък, в който остават **2 елемента - ограничения**.
* Тези елементи ще бъдат в **обхватът на елементите, които трябва да отпечатате**.
* Минете през крайния списък и **вземете само елементите, които отговарят на ограниченията.**
* Отпечатайте елементите **сортирани по азбучен ред,** разделени с интервал.

### **Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 5 23 64 2 3 34 54 12  43 23 12 31 54 51 92 | 23 23 31 34 43 51 |
| **Коментари** | |
| След смесването на двата списъка получаваме:  1 92 5 51 23 54 64 31 2 12 3 23 34 43  Остават 54 и 12 (това са нашите ограничения) и получаваме:  51 23 31 23 34 43  Накрая отпечатваме резултата сортиран | |

## **\*Барабани**

Габси е барабанистка от Orgolt's Final Revenge. Тя е има комплект за барабани, но те имат различен произход- някои ги е купила, друга са подаръци, а трети имат различно **качество.** Всеки ден тя се упражнява с барабаните. Така тя им нанася щети и им намаля качеството. Понякога те се чупят, затова трябва да си купи нови. Помогнете ѝ да ги запази чрез подреждане.

Ще получите **спестяванията** на Габси. Това са парите, които спестява за нови барабани. На следващия ред ще получите **редица от числа**, които представляват **качеството** на барабаните.

Докато не получите командата **"Hit it again, Gabsy!",** ще получавате число : **удара** на Гаспи за всеки **барабан**, докато се упражнява. Трябва да намалите качеството им с нейната сила.

Когато барабаните **достигнат качеството 0**, те се чупят. Гаспи трябва да намери алтернатива. Трябва да купи същия модел. Качеството е **същото като на счупения барабан**. Цената се изчислява със следната формула: {качеството} \* 3. Гаспи винаги ще получава нов барабан, докато може да си го **позволи**. Ако няма достатъчно пари се **премахва** от комплекта.

Програмата спира, когато се въведе **"Hit it again, Gabsy!".** Трябва да отпечатате статуса на комплекта. На втория ред трябва да отпечатате **оставащите пари**.

### **Вход**

* На **първия ред** ще получите **спестяванията** - число с плаваща запетая.
* На **втория ред** ще получите комплекта: **редица от числа**, разделена с интервал
* Докато не получите **"Hit it again, Gabsy!"** Ще получавате **числа** - силата на Гаспи за всеки барабан.

### **Изход**

* На първия ред трябва да отпечатате качеството на **всеки барабан**, разделени с **интервали**
* На втория ред ще отпечатате **оставащите пари** в следния формат: **"Gabsy has {останали пари}lv.",** закръглено до десетичната запетая

### **Бележки**

* Качеството на всеки барабан ще бъде число в обхвата **[1, 1000]**.
* Спестяванията ще бъдат в обхвата **[0, 1000],**
* Позволена **памет: 16MB**
* Позволено **време: 100ms**

### **Примери**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| 1000.00  58 65 33  11  12  18  10  Hit it again, Gabsy! | 7 14 23  Gabsy has 901.00lv. | Комплекта от барабани е – 58 65 33.  Ден 1: нейната сила е 11 => 47 54 22;  Ден 2: нейната сила е 12 => 35 42 10;  Ден 3: нейната сила е 18 => 17 24 -8;  Третия барабан се чупи. Гаспи има пари и затова го заменя => 17 24 33  Ден 4: нейната сила е 10 => 7 14 23;  Накрая отпечатваме отпечатваме статута на барабаните и нейните спестявания. |
| 154.00  55 111 3 5 8 50  2  50  8  23  1  Hit it again, Gabsy! | 27 2 4 7  Gabsy has 10.00lv. |  |