# Упражнения: Стек и опашка

Тествайте решението в Judge: <https://judge.softuni.org/Contests/4155/03-Stack-and-Queue-More-Exercises>.

## Камион

Да предположим, че има кръг, в който има **N** на брой бензиностанции. Те са номерирани от 0 до N-1 (и двете числа включително). Имате информация съставена от **две части** за всяка една бензиностанция: (1) **количеството бензин**, което наливате, и (2) **дистанцията** до следващата бензиностанция.

Имате резервоар на камиона с неограничен обем. Можете да стартирате от **всяка** една бензиностанция. **Пресметнете** от къде да тръгнете, така че да изминете **целия кръг**. Имайте предвид, че камионът ще спре на **всяка от бензиностанциите**. Камионът се гори по **един литър** бензин на **всеки километър**.

### Вход

* На първия ред получавате стойност на **N**
* На следващите **N** реда ще получавате двойки от числа, т.е. количеството бензин, което бензиностанцията ще ви даде, и разстоянието до следващата бензиностанция

### Изход

* Цяло число, което ще бъде най-малкият индекс на бензиностанция, от която можем да започнем обиколката

### Бележки

* 1 ≤ N ≤ 1000001
* 1 ≤ Количество бензин, Дистанция≤ 1000000000

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3  1 5  10 3  3 4 | 1 |

## Прост редактор

Даден ви е празен низ. Вашата задача е да имплементирате 4 команди, които манипулират низа:

* 1 текст- **Добавя** текст в края на низа.
* 2 X - **Изтрива** последните X на брой елемента
* 3 N - **Връща N-**тия **елемент** от текста
* 4 - **Отменя** последната команда от тип 1 / 2 и връща текста в състоянието преди тази операция

### Вход

* Първо ще прочетете ***n,*** броя операции.
* На следващите ***n*** реда ще получите **името на операцията**, последвано от **стойност**, ако има такава, разделени с интервал в следния формат “{**Име на командата}** {**Стойност}”.**

### Изход

* На всяка операция от тип **3** принтирайте на нов ред върнатия символ.

### Бележки

* 1 ≤ N ≤ 105
* Дължината на текста няма да надвиши 1000000.
* Всички символи ще бъдат на английски.
* Гарантирано е, че последователността от операции за въвеждане е възможна за изпълнение.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 8  1 abc  3 3  2 3  1 xy  3 2  4  4  3 1 | c  y  a | Имаме **8 операции**. Низът първоначално е празен. На следващата операция добавяме текста **"abc"**. След това принтираме **третата** буква, която е **c**.  Изтриваме трите букви – **"abc"**.  Добавяме **"xy"** към низа. След това принтираме втората буква, която е **y**. Връщаме една стъпка назад и низът става **“”.** Отново връщаме една стъпка назад и низът става **"abc".** След това принтираме третата буква, която е **c**. |

## Ключов револвер

Нашият любим супер шпионин Сам се е върнал от мисия. Този път има по-трудна задача. Трябва да **отключи сейф**. Проблемът е, че сейфът е **заключен** от **няколко ключалки**, като всички са с **различни размери**.

Нашият герой има специално оръжие, което се нарича **ключов револвер** със специални патрони. Всеки **патрон** може да отключи **ключалка** с размер **равен или по-голям** от **патрона**. Патронът влиза в ключалката и взривява **всичко**. Сам **не знае размера** на ключалките, затова трябва да стреля по тях докато не свършат ключалките.

Може би се питате какво има в сейфа? Говори се, че заклетият враг на Сам - **Николадзе**, държи вътре своята рецепта за **грузинска ракия Чача**. Едно последно нещо - всеки патрон, който Сам изстрелва, ще му струва пари, **които ще бъдат приспаднати от заплатата му** за разузнаването.

### Вход

* На **първия ред** ще получите **цените за всеки патрон** - **число в обхват** **[0-100]**
* На **втория ред** ще получите **размера на барабана**- **число в обхват [1-5000]**
* На **третия ред** ще получите **патроните** - **числа разделени с интервал в обхват [1-100]**
* На **четвъртия ред** ще получите **ключалките** - **числа разделени с интервал в обхват [1-100]**
* На **петия ред** ще получите **заплатата му** - **число в обхват [1-100000]**

След като Сам е получил информацията и екипировката (**входа**), той **започва да стреля по ключалките** отпред назад, докато патроните минават през ключалките **отзад напред**.

Ако **патронът** е **по-малък или равен** на **сегашната ключалка** отпечатайте **“Bang!”**, след това **премахнете ключалката**. В противен случай отпечатайте **"**Ping!**"**, оставяйки ключалката **непокътната**. И в двата случая **куршумите се премахват**.

Ако Сам остане без зареден барабан на определена ключалка, отпечатайте **"**Reloading!**"** на конзолата, след продължавате да стреляте. Ако няма повече патрони, **не го** отпечатвайте.

Програмата спира, когато Сам **няма повече патрони** или сейфът **няма повече ключалки**.

### Изход

* Ако Сам няма повече патрони преди ключалките на сейфът да свършат, отпечатайте:

**"**Couldn't get through. Locks left: {броя на оставащите ключалки}**"**

* Ако Сам **успее да** отвори сейфът, отпечатайте:  
  **"**{броя на оставащите ключалки коршуми} bullets left. Earned ${спечелините пари}**"**

Не забравяйте да вземете в **предвид цената на куршумите**, когато изчислявате **спечелените пари**.

### Бележки

* Входът **винаги ще бъде валиден**. **Няма нужда** да го проверявате изрично.
* Никога **няма** да има случай, в който Сам разбива ключалката и завършва с **отрицателен баланс**.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 50  2  11 10 5 11 10 20  15 13 16  1500 | Ping!  Bang!  Reloading!  Bang!  Bang!  Reloading!  2 bullets left. Earned $1300 | 20 стреля ключалка 15 (ping)  10 стреля ключалка 15 (bang)  11 стреля ключалка 13 (bang)  5 стреля ключалка 16 (bang)  Цената на коршумите:  4 \* 50 = $200  Спечелени:  1500 – 200 = $1300 |
| 20  6  14 13 12 11 10 5  13 3 11 10  800 | Bang!  Ping!  Ping!  Ping!  Ping!  Ping!  Couldn't get through. Locks left: 3 | 5 стреля ключалка 13 (bang)  10 стреля ключалка 3 (ping)  11 стреля ключалка 3 (ping)  12 стреля ключалка 3 (ping)  13 стреля ключалка 3 (ping)  14 стреля ключалка 3 (ping) |

|  |  |  |
| --- | --- | --- |
| 33  1  12 11 10  10 20 30  100 | Bang!  Reloading!  Bang!  Reloading!  Bang!  0 bullets left. Earned $1 | 10 стреля ключалка 10 (bang)  11 стреля ключалка 20 (bang)  12 стреля ключалка 30 (bang)  Цената на коршумите:  3 \* 33 = $99  Спечелени:  100 – 99 = $1 |

## \*Чаши и шишета

Ще ви бъде дадена **редица от числа** - всяко число съдържа **капацитета на чаша**. На нови ред ще получите **още една редица от числа** - обема на **шишета с вода**. Вашата задача е да **напълните** чашите.

Пълненето става чрез избиране на **точно една** бутилка. Трябва да вземете **последната получена бутилка** и да започнете да пълните от **първата въведена чаша**. Ако бутилката има **N** количество вода, **намалете** свободното количество на първата чаша с **N**.

Когато **количеството** на чашата достигне **0 или по-малко**, тя се **премахва**. **Възможно** е количеството на чашата да е **по-голямо** от количеството в бутилката. **В такъв случай** взимаме друга бутилка, докато **свободното количество** на чашата достигне 0 или по-малко. Ако обемът на бутилката е **по-голям или равен** на този на чашата, трябва да напълните чашата и останалата вода **се губи**. Трябва да **следите изразходваните литри вода** и да ги **отпечатате** в края на програмата.

Ако **сте успели да напълните всички чаши**, отпечатайте **останалите бутилки с вода**, от **последната въведена до първата**, в противен случай трябва да отпечатате **останалите чаши** – **от първата до последната**.

### Вход

* На **първия ред** ще получите редица от числа- **капацитета** на чашите, разделени с **интервал.**
* На **втория ред** ще получите редица от числа – бутилките, напълнени с вода, разделени с **интервал**.

### Изход

* На първия ред трябва да отпечатате бутилките или оставащите чаши в зависимост от случая. **Спазвайте реда** за **печат точно както е посочено**.
  + **"Bottles: {oставащи бутилки}"** или **"Cups: {оставащи чаши}"**
* На втория ред принтирайте изхабените литри от вода в следния формат:**"Wasted litters of water: {общото количество вода изхабено}".**

### Бележки

* Всички числа ще бъдат в обхвата [1, 500].
* **НЯМА** да има случай, в който водата да е точно толкова, колкото са стойностите на чашите, така че накрая да няма чаши и вода в бутилките

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 4 2 10 5  3 15 15 11 6 | Bottles: 3  Wasted litters of water: 26 | Вземеме първата чаша и последната бутилка, както е описано в условието.  6 – 4 = 2 – имаме още 2, така че похабената вода става 2.  11 – 2 = 9 – отново е повече, така че го добавяме към предишната сума, която е 2 и става 11.  15 – 10 = 5 – изразходваната вода става 16.  15 – 5 = 10 – изразходваната вода става 26.  Успяхме да напълним всички чаши, така че отпечатваме останалите бутилки и общото количество изхабена вода. |
| 1 5 28 1 4  3 18 1 9 30 4 5 | Cups: 4  Wasted litters of water: 35 |  |
| 10 20 30 40 50  20 11 | Cups: 30 40 50  Wasted litters of water: 1 |  |