# Упражнения: Масиви

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/3169/Arrays>.

## Ден от седмицата

Въведете **ден от седмицата** [1…7] и отпечатайте **името** на деня (на английски) или **"Invalid day!"**

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 | Monday |
| 2 | Wednesday |
| 10 | Invalid day! |

## Отпечатване на числа в обратен ред

Прочетете **n** на брой числа и ги принтирайте в **обратен ред.**

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 3  10  20  30 | 30 20 10 | 1  10 | 10 |

### Насоки

Първо трябва да прочетем **n** от конзолата

Graphical user interface, text, application

Description automatically generated

След това **създаваме масив** от числа с размер **n**.

Graphical user interface, text, application

Description automatically generated

Четем **n** числата чрез for-цикъл.

Graphical user interface, text, application

Description automatically generated

**Задаваме** стойност на елементите на масива.

Graphical user interface, text, application

Description automatically generated

Принтираме масива в **обратен ред**.

Graphical user interface, text, application, email

Description automatically generated

## Закръглени числа

Прочетете масив от реални числа (разделени с интервали), закръглете ги до най-близкото число и принтирайте числата.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 0.9 1.5 2.4 2.5 3.14 | 0.9 => 1  1.5 => 2  2.4 => 2  2.5 => 3  3.14 => 3 |
| -5.01 -1.599 -2.5 -1.50 0 | -5.01 => -5  -1.599 => -2  -2.5 => -3  -1.50 => -2  0 => 0 |

## Обърнат масив от низове

Прочетете **масив от низове** (разделени с **интервал**), **обърнете** го и **принтирайте** елементите:

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| a b c d e | e d c b a |
| -1 hi ho w | w ho hi -1 |

## Влак

Ще ви бъде даден броят на вагоните **n**. На следващите **n** редове ще бъде посочен **броя на хората** във **всеки вагон**. Отпечатайте **броя на хората във всеки вагон** и на **следващият ред,** **общият брой на хората** във влака.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3  13  24  8 | 13 24 8  45 |
| 6  3  52  71  13  65  4 | 3 52 71 13 65 4  208 |
| 1  100 | 100  100 |

## Зиг-Заг масиви

Напишете програма, която създава **2 масива**. Ще ви бъде даден число **n**. На следващите **n** редове ще получавате по **2 числа.** Образувайте **2 масива**, както е показано в примерите:

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4  1 5  9 10  31 81  41 20 | 1 10 31 20  5 9 81 41 |
| 2  80 23  31 19 | 80 19  23 31 |

## ---------- TODO: MOVE TO ADVANCED

## Завъртане на масиви

Напишете програма, която получава **масив** и **брой ротации**, които трябва да изпълните (първият елемент отива накрая). Отпечатайте получения масив.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 51 47 32 61 21  2 | 32 61 21 51 47 |
| 32 21 61 1  4 | 32 21 61 1 |
| 2 4 15 31  5 | 4 15 31 2 |

## Топ числа

Напишете програма, която **намира топ числата** в масива. Едно число е **топ**, ако е **по-голямо** от всички елементи **от дясната му страна**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 4 3 2 | 4 3 2 |
| 14 24 3 19 15 17 | 24 19 17 |
| 27 19 42 2 13 45 48 | 48 |

## Еднакви суми

Напишете програма, която определя дали **съществува елемент в масива**, така че **сумата на елементите отляво** да е **равна** на **сумата на елементите отдясно (може да има не само един 1 такъв елемент)**. Ако **няма елементи от ляво/дясно**, тяхната **сума се зачита за 0**. Отпечатайте **индекса**, който отговаря на условието, или "**no**" ако няма такъв индекс.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 1 2 3 3 | 2 | В a[2] -> лява сума = 3, дясна сума = 3  a[0] + a[1] = a[3] |
| 1 2 | no | В a[0] -> лява сума = 0, дясна сума = 2  В a[1] -> лява сума = 1, дясна сума = 0  Не съществува такъв индекс |
| 1 | 0 | В a[0] -> лява сума = 0, дясна сума = 0 |
| 1 2 3 | no | Не съществува такъв индекс |
| 10 5 5 99 3 4 2 5 1 1 4 | 3 | В a[3] -> лява сума = 20, дясна сума = 20  a[0] + a[1] + a[2] = a[4] + a[5] + a[6] + a[7] + a[8] + a[9] + a[10] |

## Максимална последователност от равни елементи

Напишете програма, която **намира най-дългата еднаква редица** от елементи в масива от числа. Ако има няколко такива редици, отпечатайте най-лявата.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 2 1 1 2 3 3 **2 2 2** 1 | 2 2 2 |
| **1 1 1** 2 3 1 3 3 | 1 1 1 |
| **4 4 4 4** | 4 4 4 4 |
| 0 **1 1** 5 2 2 6 3 3 | 1 1 |

## \*Фабрика Камино

Клониращата фабрика в Камино има поръчка за клониране на войски. Вашата задача е да намерите **най-добрата ДНК** последователност, която да се използва за продукцията.

Ще получите **ДНК дължина** докато не получите командата **"Clone them!"**. Ще ви бъде дадена **ДНК редица от единици и нули, разделени чрез "!" (един или няколко).**

Вие трябва да изберете **последователността с най-дългата редица**. Ако има няколко редици с **еднаква дължина на последователност от единици**, отпечатайте тази с **най-левия начален индекс**, ако има няколко дължини с еднаква **дължина и еднакъв първоначален индекс**, изберете подредицата с **най-голямата сума** от нейните елементи.

След като получите последната команда **"Clone them!"** трябва да отпечатате събраната информация в следния формат:

"Best DNA sample {най-добърият индекс на редицата} with sum:{най-добърата сума на редицата}."

"{ДНК редицата, разделена по празно място}"

### Вход

* **Първият ред** съдържа **дължината** на **редиците-число в обхвата [1...100];**
* На следващите редове докато не получите **"Clone them!",** ще получавате редеци (поне една) от единици и нули **разделени чрез "!"** (един или няколко).

### Изход

Изхода трябва да отпечатан на конзолата и да съдържа два реда със следния формат:

"Best DNA sample {редицата с най-добър индекс} with sum: {редицата с най-голяма сума}"

"{ДНК редицата, разделена с интервал}"

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 5  1!0!**1!1**!0  0!**1!1**!0!0  Clone them! | Best DNA sample 2 with sum: 2.  0 1 1 0 0 | Получаваме 2 редици с **еднаква дължина на подредици от единици**, но се отпечатва втората защото подредицата стартира от **индекс[1].** |
| **Вход** | **Изход** | **Обяснения** |
| 4  **1!1**!0!**1**  1!0!0!1  **1!1**!0!0  Clone them! | Best DNA sample 1 with sum: 3.  1 1 0 1 | Получаваме 3 редици. Първата и третата **имат еднаква дължина** от подредици от единици ->, **и двете стартират от индекс[0],** но се принтира първата, защото й **сумата е по-голяма**. |

## \*Калинки

Дадено ви е **поле с размер** и **индекси на калинки** в полето. След това на всеки нов ред **до подаване на командата "end"**, **калинката променя** позицията **наляво** или **надясно** с **дадена дължина**.

**Командата на калинката** изглежда така: **"0 right 1"**. Това означава, че малкото насекомо е на индекс 0 и трябва да прелети един индекс надясно. Ако калинката **кацне на друга калинка**, тя **продължава да лети** в същата директория **със същата посока**. Ако калинката **излети извън полето, тя изчезва**.

Например, представете си, че ви е дадено поле с размер 3 и има калинка на индекс 0 и 1. Ако калинката на индекс 0 трябва да прелети надясно с дължина 1 (0 right 1) ще се опита да кацне на индекс 1, но тъй като там има друга калинка, тя ще продължи по-надясно с допълнителна дължина 1, кацайки на индекс 2. След това, ако същата калинка трябва да лети надясно с дължина 1 (2 right 1), тя ще кацне някъде извън полето, така че отлита:



Ако Ви бъде даден индекс, на който няма калинка, нищо не се случва. Ако Ви бъде даден индекс на калинка извън полето, нищо не се случва.

Вашата работа е да напишете програма, която симулира летенето на калинките. Накрая **отпечатвайте всички клетки в полето, разделени с интервал**. За всяка клетка, в която има калинка, отпечатайте „**1**“, а за всяка празна клетка отпечатайте „**0**“. За примера по-горе изходът трябва да бъде '**0 1 0**'.

### Вход

* На първият вход ще получавате число - размерът на полето
* На вторият ред ще получавате **индексите** на калинките, разделени с празно място. **Индексите могат** да бъдат или да не бъдат в полето
* На следващите редове докато не получите командата “end”, вие получавате команди в следния формат: "**{индекс на калинката} {директория} {дължина на летене}**"
* Отпечатайте всички клетки със следния формат: **"{клетка} {клетка} … {клетка}"**
  + Ако в клетка има калинка, отпечатайте **'1'**
  + Ако клетка е празна, отпечатайте **'0'**

### Бележки

* Размерът в полето варира в обхвата [0 … 1000]
* Индексите на калинките ще бъдат в обхавата [-2,147,483,647… 2,147,483,647]
* Номерът на командите ще бъдат в обхвата [0 … 100]
* Дължината на полетите ще бъде в обхвата [-2,147,483,647… 2,147,483,647]

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 3  0 1  0 right 1  2 right 1  end | 0 1 0 | 1 1 0 - първоначално поле  0 1 1 - полето след "0 right 1"  0 1 0 - полето след "2 right 1" |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| 3  0 1 2  0 right 1  1 right 1  2 right 1  end | 0 0 0 |  | 5  3  3 left 2  1 left -2  end | 0 0 0 1 0 |