# Упражнения: Масиви

Тествайте решението в Judge: <https://judge.softuni.org/Contests/4144/01-1-Arrays-Basics>.

## 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 |

## Умножение на числа

Напишете програма, която чете от конзолата цяло число **N** и създава **масив** от **N елемента**, като всеки елемент е **равен на индекса си**, **умножен** по **5**. Отпечатайте елементите на масива на конзолата на **нов ред**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 4 | 0, 5, 10, 15 |

## Еднаквост

Напишете програма, която чете от конзолата **два масива** от цели числа и проверява дали са **еднакви**. Ако са еднакви, отпечатайте **"Yes"**. В противен случай, отпечатайте **"No"**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3  1 2 3 | Yes |
| 1 2 3  1 3 2 | No |

## Цифри от индекс

Напишете програма, която чете от конзолата цяло число **N** и създава **масив** от **N** елемента, като всеки елемент е равен на **сумата от цифрите на индекса** си. Отпечатайте елементите на масива на конзолата, разделени с **нов ред**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5 | 0 1 2 3 4 |
| 11 | 0 1 2 3 5 6 7 8 9 1 |

## 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 |

## Подмножества

Напишете програма, която чете от конзолата **два масива** от цели числа и проверява дали вторият е **подмножество** на **първия**, т.е. **всички елементи** на **втория** масив се съдържат в **първия**. Ако е подмножество, отпечатайте на конзолата "**Yes**". Ако не, отпечатайте "**No**".

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 4 5 6 7 8 9 10  1 2 3 | Yes |
| 1 2 3 4 5 6 7 8 9 10  -1 1 2 3 | No |

## 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 |

1. Сума на всички елементи в масив: Напишете програма, която намира и извежда сумата на всички елементи в целочислен масив.
2. Намиране на средната стойност: Създайте масив с числа и намерете средната стойност (аритметично средно) на всички елементи.
3. Обръщане на масив: Напишете програма, която обръща реда на елементите в масива. Например, ако имате масив [1, 2, 3, 4, 5], програмата трябва да върне [5, 4, 3, 2, 1]
4. Проверка за съществуване на даден елемент: Напишете програма, която приема число и проверява дали то се среща в даден масив. Ако елементът се среща, изведете съобщение, че той съществува в масива.
5. Проверка за палиндром: Напишете програма, която проверява дали даден масив от символи е палиндром. Палиндром е дума или фраза, която се чете по един и същ начин отпред назад и обратно.
6. Извлечение на уникални елементи: Напишете програма, която извлича уникалните елементи от масива и ги извежда, премахвайки дубликатите. Например, ако имате масив [1, 2, 2, 3, 4, 4, 5], програмата трябва да изведе [1, 2, 3, 4, 5].
7. Обединяване на два масива: Напишете програма, която обединява два масива в трети масив и извежда резултата.
8. Подреждане на четни и нечетни числа: Напишете програма, която подрежда четните числа от даден масив преди нечетните. Редът на четните и нечетните числа в резултатния масив е неважен.
9. Брой срещания на даден елемент: Напишете програма, която намира и извежда броя на срещанията на даден елемент в масива. Задайте елементът, който искате да проверите за срещания.