# Упражнения: Масиви

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

## Ден от седмицата

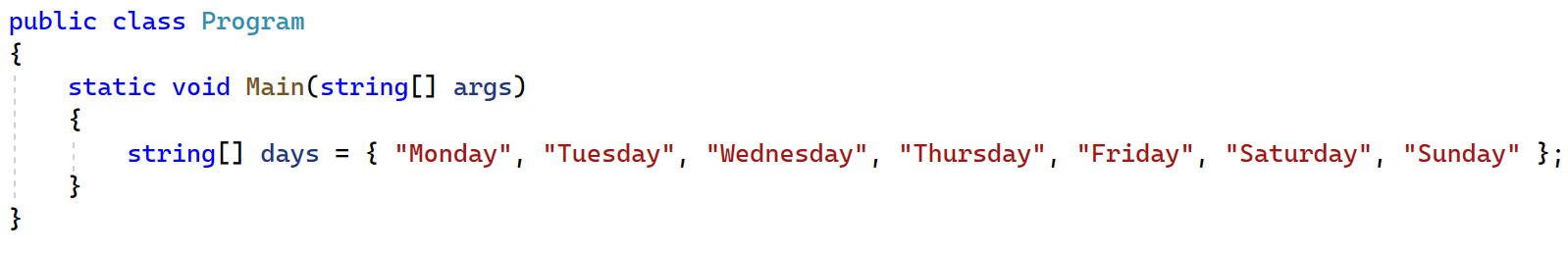
Въведете **ден от седмицата** [1…7] и отпечатайте **името** на деня (на английски) или **"Invalid day!"**

### Примери

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

### Насоки

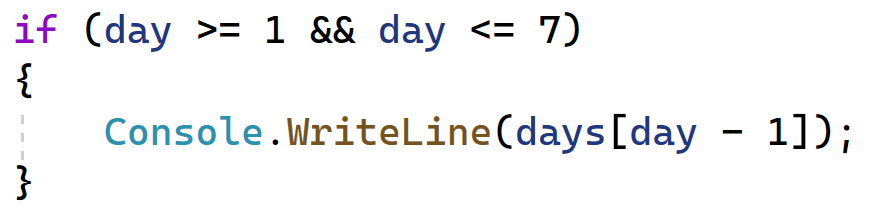
1. Първо създайте **масива** за **дните** от **седмицата**:



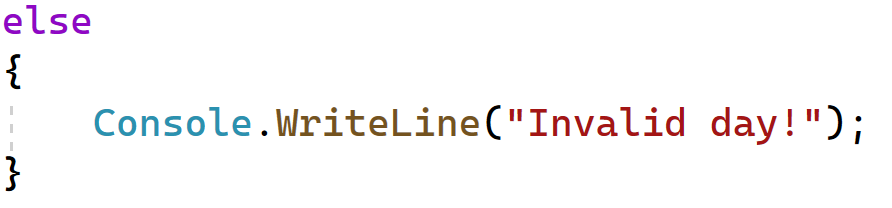
1. След това прочитет числото **day** от конзолата:



1. Създайте проверка дали **day** е **по-голямо или равно** на **1** и **по-малко или равно** на **7**. Ако това е така, отпечатайте деня от седмицата:



1. В противен случай, отпечатайте **" Invalid day!"**:



## Отпечатване на числа в обратен ред

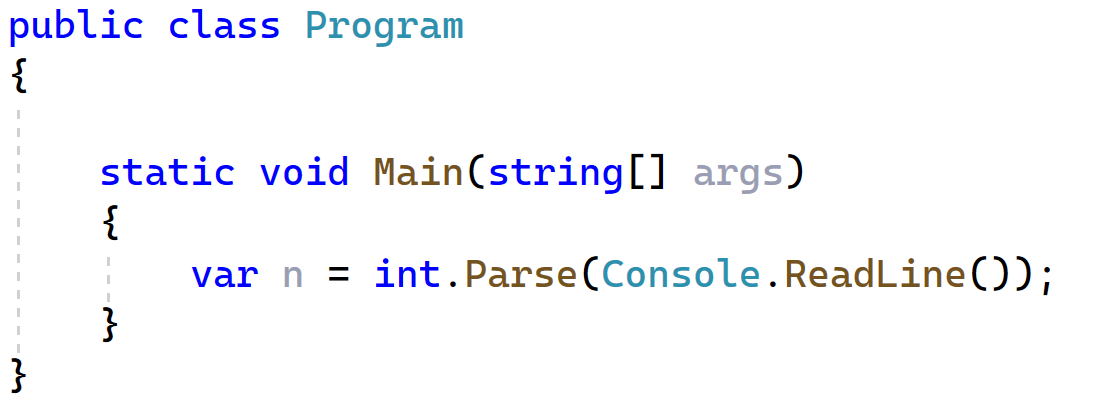
Прочетете **n** на брой числа и ги принтирайте в **обратен ред.**

### Примери

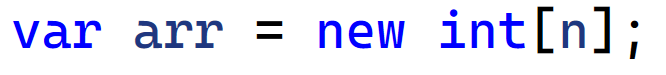
|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 3  10  20  30 | 30 20 10 | 1  10 | 10 |

### Насоки

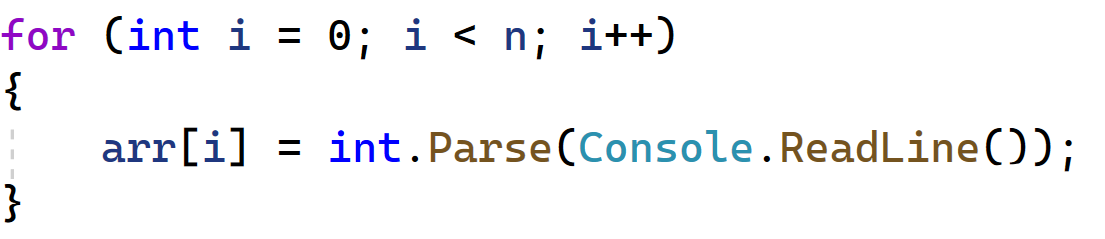
1. Първо прочетете **n** от конзолата:



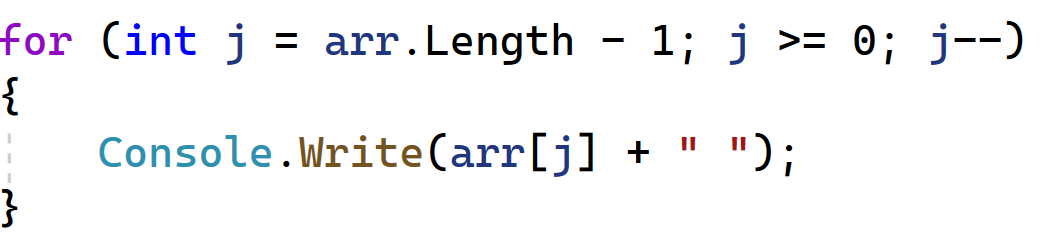
1. След това **създавайте масив** от числа с размер n:



1. Прочете числата на масива чрез **for-цикъл**:



1. Принтирайте масива в **обратен ред**:



## Закръглени числа

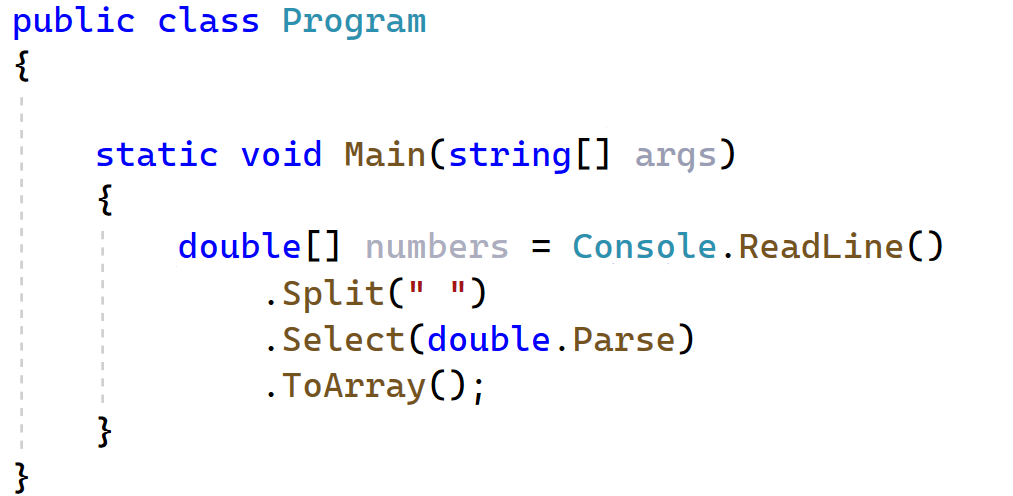
Прочетете масив от реални числа (разделени с интервали), закръглете ги до най-близкото число и принтирайте числата.

### 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 |

### Насоки

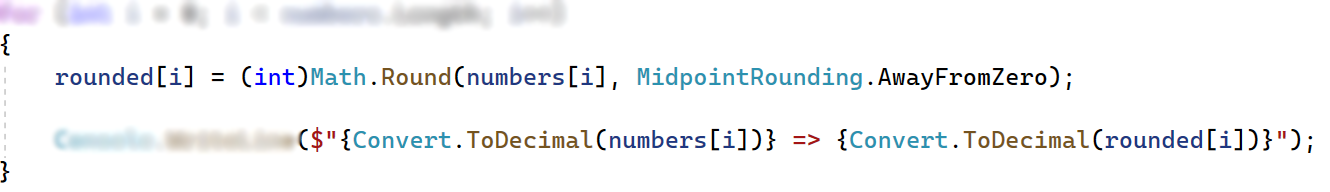
1. Създайте масив **numbers**.
2. **Прочетете** масива чрез **System.LINQ**:



1. Създайте нов масив rounded, който има дължината на масива numbers:



1. След това създайте **for-цикъл**,който ще обхожда до дължината на масива numbers.
2. В него присвояваме **закръгленото** число чрез метода **Math.Round()** на масива **rounded с индекс** iи отпечатваме **изхода**:

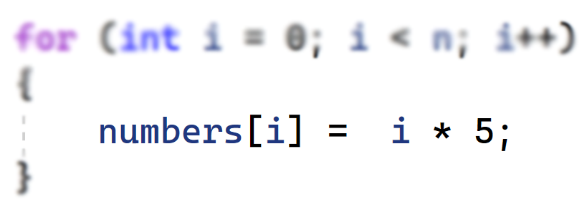


## Умножение на числа

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**.

### Насоки

* Създайте променлива **n** и масив **numbers** с дължина **n**
* Създайте **for-цикъл**
* В него задайте на масива **numbers** с **текущ индекс i** стойност **i**\*5



* Накрая отпечатайте умножените числа по 5

### 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"**.

### Насоки

* Създайте два масива с име **arr1** и **arr2**
* Наследяващия ред създайте променлива **areEqual** със стойност **true**
* Създайте **for-цикъл**, чрез който програмата ще сравнява всеки елемент на **arr1** със съответния елемент на **arr2**.
* Ако има елементи, които не съвпадат, трябва да отпечатаме **"No"**, след това да променим стойността на **areEqual** на **false** и да излезем от цикъла, за да оптимизираме изпълнението на програмата
* Ако всички елементи на **arr1** съвпадат със съответните елементи на **arr2**, трябва да изпишем **"Yes"**

### 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  4  5  6  7  8  9  1 |

### Насоки

* Започнете с въвеждане на числото **n**
* Създайте масив **arr** с размер **n**
* Направете **for-цикъл** от **0** до **n-1**
* Създайте променлива **index** със стойност **i**
* След това изчислете колко цифри се съдържат в текущата променлива **index**
* Запазете резултата в **arr[i]**
* Накрая отпечатайте всички елементи на масива **arr**

## 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 |