# Упражнение: Рекурсия

Тествайте решението в Judge: <https://judge.softuni.org/Contests/4176/15-Recursion>.

## Обърнат масив

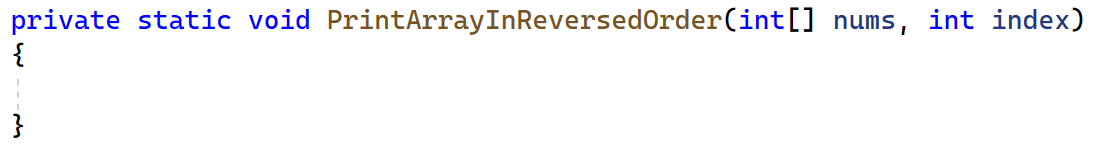
Напишете програма, **която отпечатва масива в обратния ред**, чрез **рекурсия**.

### Примери

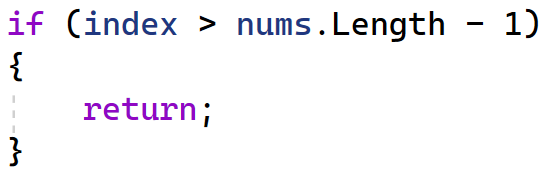
|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 4 5 6 | 6 5 4 3 2 1 |

### Насоки

1. Напишете рекурсивен метод, който като **аргументи** приема **масив** и **индекс**:



1. В него трябва да напишете дъното. Това е когато **сегашният** **индекс** е **по-голям** от **дължината** на **масива -1**:



1. След това извикайте **метода** като трябва да подадете **следващия** **индекс**:



1. Накрая отпечатайте числото:



## Рекурсивна сума на масива

Напишете програма, която намира сума на всички елементи в масив от числа. Използвайте **рекурсия**.

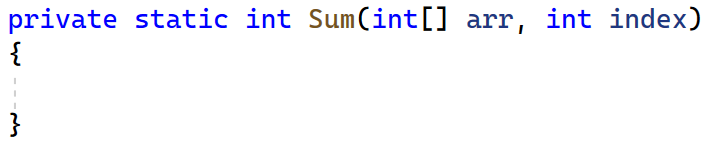
**Бележка**: В практиката не трябва да се използва рекурсия (вместо това използвайте **итеративно решение**). Това е само за упражнение

### Примери

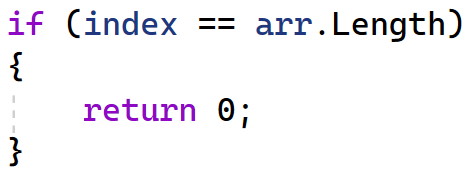
|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 4 | 10 |
| -1 0 1 | 0 |

### Насоки

1. Напишете рекурсивен метод, който като аргументи трябва да приема **масив** и **индекс**:



1. В него трябва да напишете дъното. Това е когато **няма повече елементи** в масива:



1. Създайте променлива, в която да запазите стойността на **сегашния** **елемент** + **сумата** от **следващите** **елементи** (получена чрез рекурсивно извикване):



1. Накрая върнете стойността на променливата:



## Рекурсивен факториел

Напишете програма, която намира факториел на дадено число. Използвайте **рекурсия**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5 | 120 |
| 10 | 3628800 |

### Насоки

1. Напишете **рекурсивен** метод, който трябва да взема като аргумент число **n**.
2. Методът трябва да върне **сегашният елемент** \* **резултатът от пресмятането на факториел от сегашният елемент - 1**.
3. Рекурсията трябва да **спре**, когато няма **повече елементи** в **масива**:

A picture containing text

Description automatically generated

1. Горният начин на решаване ще бъде **бавен**, когато **n** e голямо число. Можете да съкратите изчисленията като запомните всяка **стойност** в **масив**, която вече е **изчислена**.

## Рекурсивно рисуване

Напишете програма, която рисува фигури зависимост от n. Използвайте **рекурсия**.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 2 | \*\*  \*  #  ## |
| 5 | \*\*\*\*\*  \*\*\*\*  \*\*\*  \*\*  \*  #  ##  ###  ####  ##### |

### Насоки

1. Напишете **рекурсивен** метод, който трябва да взема като аргумент число **index**.
2. Задайте **дъно** на **рекурсията**:

Graphical user interface

Description automatically generated with low confidence

1. Отпечатайте символите **"\*"** **извиканото** преди **рекурсията** и отпечатайте **"#"** след **извиканото** на **рекурсията**:

Graphical user interface, text

Description automatically generated

## Вложени цикли към рекурсия

Напишете програма, която **симулира изпълнението на n вложени цикъла от 1 до n**. Програмата отпечатва стойностите на всички променливи във всеки момент на един ред. **Използвайте рекурсия.**

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Решение с вложени цикли (приема се, че n е положително число)** |
| 2 | 1 1  1 2  2 1  2 2 | int n = 2;  for (int i1 = 1; i1 <= n; i1++)  {  for (int i2 = 1; i2 <= n; i2++)  {  Console.WriteLine($"{i1} {i2}");  }  } |
| 3 | 1 1 1 1 1 2  1 1 3  1 2 1  1 2 2  …  3 2 3  3 3 1  3 3 2  3 3 3 | int n = 3;  for (int i1 = 1; i1 <= n; i1++)  {  for (int i2 = 1; i2 <= n; i2++)  {  for (int i3 = 1; i3 <= n; i3++)  {  Console.WriteLine($"{i1} {i2} {i3}");  }  }  } |

### Насоки

1. Прочетете за рекурсивния вложен цикъл: <https://introprogramming.info/english-intro-csharp-book/read-online/chapter-10-recursion>