**Лаб: Повторения с цикли – For-цикъл**

Задачи за упражнение в клас и за домашно към курса ["Основи на програмирането" @ СофтУни](https://softuni.bg/courses/programming-basics).

Тествайте решенията си тук: <https://judge.softuni.bg/Contests/2417>

## Числа от 1 до 100

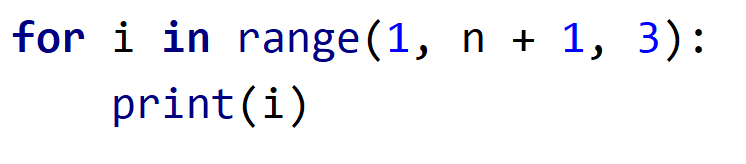
Напишете програма, която отпечатва числата от 1 до 100, всяко на нов ред.

### Примерен вход и изход

|  |  |
| --- | --- |
| **вход** | **изход** |
| (няма) | 1  2  3  …  98  99  100 |

### Насоки

Напишете решението на задачата, като си помогнете с кода от картинката по-долу:



## Числата от 1 до N през 3

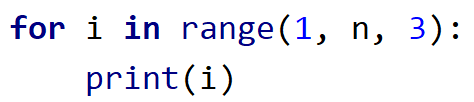
Напишете програма, която чете число n, въведено от потребителя и отпечатва **числата от 1 до** n **през 3**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 10 | 1  4  7  10 | 7 | 1  4  7 | 15 | 1  4  7  10  13 |

### Насоки

1. Прочетете входните данни от конзолата – **едно цяло число n**:
2. Направете for **цикъл** **от 1 до n** (**включително**) и задайте **стъпка 3**. Това означава, че при всяка итерация на цикъла, променливата i **ще увеличава стойността си с 3, вместо с 1**. Принтирайте променливата при всяка итерация:



## Четни степени на 2

Да се напише програма, която чете число n, въведено от потребителя, и **печата четните степени на 2** **≤** **2n**: **20**, **22**, **24**, **26**, …, **2n**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 3 | 1  4 | 4 | 1  4  16 | 5 | 1  4  16 | 6 | 1  4  16  64 | 7 | 1  4  16  64 |

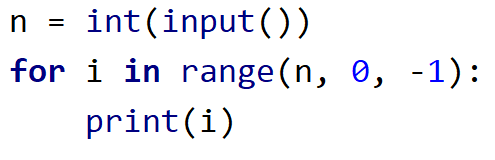
## Числата от N до 1 в обратен ред

Напишете програма, която чете цяло положително число n, въведено от потребителя и печата **числата от** n **до** **1** **в обратен ред**. Въведеното число **n**, винаги ще бъде по-голямо от 1.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | 2  1 | 3 | 3  2  1 | 5 | 5  4  3  2  1 |

### Насоки

1. Прочетете **едно цяло число** от конзолата – броят на числата, които предстои да бъдат въведени
2. Направете for **цикъл от n до 0**, като **намалявате променливата i** **с 1** при **всяка итерация** и я принтирате.
3. 

## Поток от символи

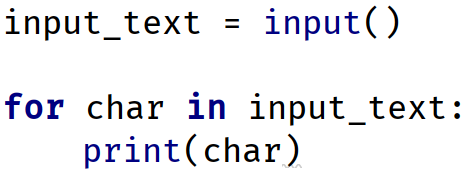
Напишете програма, която чете **текст** (стринг), въведен от потребителя и печата всеки **символ** от текста на отделен ред.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **вход** | **изход** | **вход** | **изход** |
| softuni | s  o  f  t  u  n  i | ice cream | i  c  e    c  r  e  a  m |

### Насоки

1. За да итерира **for** цикълът през въведената дума на мястото на фунцията range(), използвайте директно променливата, в която преди това сте записали входа от конзолата:



## Сумиране на гласните букви

Да се напише програма, която чете **текст** (стринг), въведен от потребителя, и изчислява и отпечатва **сумата от стойностите на гласните букви** според таблицата по-долу:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| буква | a | e | i | o | u |
| стойност | 1 | 2 | 3 | 4 | 5 |

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **коментар** |
| hello | 6 | e + o = 2 + 4 = 6 |
| hi | 3 | i = 3 |
| bamboo | 9 | a + o + o = 1 + 4 + 4 = 9 |
| beer | 4 | e + e = 2 + 2 = 4 |

## Сумиране на числа

Да се напише програма, която **чете** n-на брой **цели числа**, въведени от потребителя **и ги сумира**.

* От първия ред на входа се въвежда броят числа n.
* От следващите n реда се въвежда по едно цяло число.

Програмата трябва да прочете числата, да ги сумира и да отпечата сумата им.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2  10  20 | 30 | 3  -10  -20  -30 | -60 | 4  45  -20  7  11 | 43 | 1  999 | 999 | 0 | 0 |

## Редица цели числа

Напишете програма, която чете **n на брой цели числа**. Принтирайте **най-голямото** и **най-малкото** число сред въведените.

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| 5  10  20  304  0  50 | Max number: 304  Min number: 0 | 6  250  5  2  0  100  1000 | Max number: 1000  Min number: 0 |

## Лява и дясна сума

Да се напише програма, която чете **2 \*** n-на брой цели числа, подадени от потребителя, и проверява дали **сумата на първите** n **числа** (лява сума) е равна на **сумата на вторите** n **числа** (дясна сума). При равенство печата " Yes, sum = " + **сумата**; иначе печата " No, diff = " + **разликата**. Разликата се изчислява като положително число (по абсолютна стойност).

### Примерен вход и изход

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **коментар** |  | **вход** | **изход** | **коментар** |
| 2  10  90  60  40 | Yes, sum = 100 | 10+90 = 60+40 = 100 | 2  90  9  50  50 | No, diff = 1 | 90+9 ≠ 50+50  Difference = |99-100| = 1 |

## Четна / нечетна сума

Да се напише програма, която чете n-на брой цели числа, подадени от потребителя и проверява дали **сумата от числата на четни позиции** е равна на **сумата на числата на нечетни позиции**.

* Ако сумите са равни да се отпечатат два реда: "Yes" и на нов ред "Sum = " + **сумата**;
* Ако сумите не са равни да се отпечат два реда: "No" и на нов ред "Diff = " + **разликата**.

Разликата се изчислява по **абсолютна стойност**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **коментар** |  | **вход** | **изход** | **коментар** |  | **вход** | **изход** | **коментар** |
| 4  10  50  60  20 | Yes  Sum = 70 | 10+60 = 50+20 = 70 | 4  3  5  1  -2 | No  Diff = 1 | 3+1 ≠ 5-2  Diff = |4-3| = 1 | 3  5  8  1 | No  Diff = 2 | 5+1 ≠ 8  Diff = |6-8| = 2 |