# 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/Compete/Index/2322>

## 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

1. Създайте **нов файл с име** "numbers1To100".
2. Отидете в тялото на функцията и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



## Числата от N до 1 в обратен ред

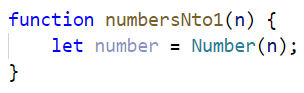
Напишете програма, която чете цяло положително число n, въведено от потребителя, и печата **числата от** n **до** **1** **в обратен ред** (от най-голямото към най-малкото).

### 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. Създайте функция с един параметър(число)**:**



1. Направете **един for цикъл от** n **до 0**, но **вместо да увеличавате** **променливата** i с 1, я **намаляйте** **с 1** при **всяка итерация на цикъла**:



1. **В тялото на цикъла** **принтирайте променливата** i:



## Числата от 1 до N през 3

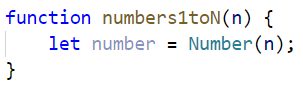
Напишете програма, която чете число n, въведено от потребителя, и отпечатва **числата от 1 до** n **през 3** (със стъпка 3).

### 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. Създайте функция с един параметър(число)**:**



1. 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:**

 **В тялото на цикъла** **принтирайте променливата** **i**:



## 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 |

## 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 |

## 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 |

## Сбор от букви

Напишете програма, която изчислява дали бюджетът ни е достатъчен, за да си купим продукта, който искаме. В името на продукта всяка гласна буква е със стойност 3, а всяка съгласна със стойност 1. Цената на продукта се изчислява като сумата от стойностите на буквите се умножи по дадено контролно число. За гласни букви се считат: "a", "e", "i", "o", "u" и "y".

### Вход

Програмата прочита **3 стойности**, въведени от потребителя:

* **име на продукта** - **текст (малки букви)**
* **контролно число** - **число** в интервала **[1.0...50.0]**
* **бюджет** - **цяло** **число** в интервала **[0...99999999]**

### Изход

Ако бюджетът е по-малък от **стойността на продукта**, тогава се отпечатва съобщението:

* **"Cannot buy {име на продукта}. Product value: {стойност на продукта}"**

В противен случай се отпечатва съобщението:

* **"{име на продукта} bought. Money left: {останали пари}"**

Всички стойности да се закръглят до втория знак след десетичната запетая.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **пример** |
| apple  2  20 | apple bought. Money left: 2.00 | В "**apple**" има 2 гласни и 3 съгласни  За всяка гласна стойността се увеличава с 3, а за съгласна с 1;  Общо: 2 \* 3 + 3 \* 1 = 9  След умножаване с контролното число получаваме стойността: 9 \* 2 = 18  Ако бюджета е по голям или равен на стойността принтираме: **"{име на продукта} bought. Money left: {останали пари}**  Следователно останалите пари са равни на: 20 – 18 = 2 и на конзолата се принтира съобщението:  **"apple bought. Money left: 2.00"** |
| milk  1.4  8 | Cannot buy milk. Product value: 8.40 | В "**milk**" има 1 гласна и 3 съгласни;  Стойност: (3 \* 1 + 1 \* 3) \* 1.4 = 8.40 |

## **Специални числа**

Напишете програма, която чете три числа – **start**, **end** и **n**. Програмата трябва да отпечатва на нови редове всички числа от **start** до **end** (включително), като всяко число, което се дели на **n** (без остатък) трябва да бъде отпечатано по следния начин: "**Special number:** **{съответното число}**" , а всяко число, което се дели на **n2**(без остатък), трябва да бъде отпечатано по следния начин: "**Very special number: {съответното число}** ".

### Вход

Програмата прочита **3 числа**, въведени от потребителя:

* **start** - **цяло число** в интервала **[0...2000]**
* **end** - **цяло** **число** в интервала **[0...2000]**
* **n** - **цяло** **число** в интервала **[0...2000]**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **описание** |
| 1  25  3 | 1  2  Special number: 3  4  5  Special number: 6  7  8  Very special number: 9  10  …  Special number: 21  22  23  Special number: 24  25 | Първото число е 1, 1%3 не е равно на 0. Следователно това число се принтира. Следващото е 2, повтаря се същото нещо. Идва ред на 3, 3%3 е равно на 0. Следователно се принтира по следния начин: "**Special number:** **3**". И така продължава докато не дойде ред на числото 9, 9%3 = 0 и 9%3**2** = 0, следователно се принтира по следния начин: "**Very special number: 9**" и т.н. |
| 1  10  4 | 1  2  3  Special number: 4  5  6  7  Special number: 8  9  10 |  |

## Число на трета степен

Напишете програма, която чете едно цяло число – **n**. В зависимост от това дали **n** е **четно или нечетно**, програмата трябва да отпечатва за всяка **четна или нечетна стойност от 1 до n** (включително) следния ред: **"Current number: A. Cube: B"** , където **A** е текущото число, а **B** е текущото число, **повдигнато на трета степен**.

### Вход

Програмата прочита **1 числo**, въведенo от потребителя:

* **n** - **цяло число** в интервала **[1...50]**

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **вход** | **изход** | **описание** |
| 5 | Current number: 1. Cube: 1  Current number: 3. Cube: 27  Current number: 5. Cube: 125 | Прочитаме числото **5** което е нечетно. Следователно ще трябва да принтираме числата **1, 3, 5**. Първото число е **1**. **"Current number: A. Cube: B"** , където **A** е текущото число **1**, а **B** е текущото число, **повдигнато на трета степен** **1,** следователно принтираме: **"Current number: 1. Cube: 1".**  Следващото число е **3**. **3** на трета степен е **27** следователно принтираме:  **"Current number: 3. Cube: 27".**  Следващото число е 5. Принтираме:  **"Current number: 5. Cube: 125".** |
| 6 | Current number: 2. Cube: 8  Current number: 4. Cube: 64  Current number: 6. Cube: 216 | Прочитаме числото **6** което е нечетно. Следователно ще трябва да принтираме числата **2, 4, 6**. Първото число е **2**. **"Current number: A. Cube: B"** , където **A** е текущото число **2**, а **B** е текущото число, **повдигнато на трета степен** **8,** следователно принтираме: **"Current number: 2. Cube: 8".**  Следващото число е **4**. **4** на трета степен е **64** следователно принтираме:  **"Current number: 4. Cube: 64".**  Следващото число е 5. Принтираме:  **"Current number: 6. Cube: 216".** |