

## Лаб: Повторения с цикли – For-цикъл

### 1. 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<br>2<br>3<br>...<br>98<br>99<br>100 |

#### Насоки

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

```
for number in range(1, 101):  
    print(number)
```

### 2. 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<br>4<br>7<br>10 | 7    | 1<br>4<br>7 | 15   | 1<br>4<br>7<br>10<br>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:

```
for number in range(1, n + 1, 3):  
    print(number)
```

### 3. Четни степени на 2

Да се напише програма, която чете число **n**, въведено от потребителя, и печата четните степени на 2  $2 \leq 2^n$ :  $2^0$ ,  $2^2$ ,  $2^4$ ,  $2^6$ , ...,  $2^n$ .

#### Примерен вход и изход

| вход | изход  |
|------|--------|
| 3    | 1<br>4 |

| вход | изход        |
|------|--------------|
| 4    | 1<br>4<br>16 |

| вход | изход        |
|------|--------------|
| 5    | 1<br>4<br>16 |

| вход | изход              |
|------|--------------------|
| 6    | 1<br>4<br>16<br>64 |

| вход | изход              |
|------|--------------------|
| 7    | 1<br>4<br>16<br>64 |

### 4. 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  |
|------|--------|
| 2    | 2<br>1 |

| вход | изход       |
|------|-------------|
| 3    | 3<br>2<br>1 |

| вход | изход                 |
|------|-----------------------|
| 5    | 5<br>4<br>3<br>2<br>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**.

```
n = int(input())
for i in range(n, 0, -1):
    print(i)
```

3.

### 5. Поток от символи

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

#### Примерен вход и изход

| вход    | изход                           | вход      | изход                               |
|---------|---------------------------------|-----------|-------------------------------------|
| softuni | s<br>o<br>f<br>t<br>u<br>n<br>i | ice cream | i<br>c<br>e<br><br>c<br>r<br>e<br>a |

|  |  |  |   |
|--|--|--|---|
|  |  |  | m |
|--|--|--|---|

## Насоки

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

```
input_text = input()

for char in input_text:
    print(char)
```

## 6. Сумиране на гласните букви

Да се напише програма, която чете **текст** (string), 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     | $e + e = 2 + 2 = 4$         |

## 7. 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    | 30    | 3    | -60   | 4    | 43    | 1    | 999   | 0    | 0     |
| 10   |       | -10  |       | 45   |       | 999  |       |      |       |
| 20   |       | -20  |       | -20  |       |      |       |      |       |
|      |       | -30  |       | 7    |       |      |       |      |       |
|      |       |      |       | 11   |       |      |       |      |       |

## 8. 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    | Max number: 304 | 6    | Max number: 1000 |
| 10   | Min number: 0   | 250  | Min number: 0    |
| 20   |                 | 5    |                  |
| 304  |                 | 2    |                  |
| 0    |                 | 0    |                  |
| 50   |                 | 100  |                  |
|      |                 | 1000 |                  |

## 9. Лева 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** (лева 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    | Yes, sum = 100 | 10+90 = 60+40 = 100 | 2    | No, diff = 1 | 90+9 ≠ 50+50 |
| 10   |                |                     | 90   |              | Difference = |
| 90   |                |                     | 9    |              | 99-100  = 1  |
| 60   |                |                     | 50   |              |              |
| 40   |                |                     | 50   |              |              |

## 10. 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 отпечата два реда: "Yes" и на нов ред "Sum = " + сумата;
- Ако сумите не са равни да се отпечата 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    | Yes      | 10+60 =  | 4    | No       | 3+1 ≠ 5-2 | 3    | No       | 5+1 ≠ 8  |
| 10   | Sum = 70 | 50+20 =  | 3    | Diff = 1 | Diff =    | 5    | Diff = 2 | Diff =   |
| 50   |          | 70       | 5    |          | 4-3  = 1  | 8    |          | 6-8  = 2 |
| 60   |          |          | 1    |          |           | 1    |          |          |
| 20   |          |          | -2   |          |           |      |          |          |