

# Лаб: Повторения с цикли – while-цикъл

Задачи за упражнение и домашно към курса "[Основи на програмирането](#)" в СофтУни.

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1014>.

## 1. Число в диапазона [1...100]

Напишете програма, която въвежда цяло положително число **n** в диапазона [1...100]. При въвеждане на число извън посочения диапазон да се отпечата съобщение за грешка и **да се въведе ново число**.

### Примерен вход и изход

Вход / Изход
35 The number is: 35
105 Invalid number!
0 Invalid number!
-200 Invalid number!
77 The number is: 77

### Насоки

1. Създайте **нов проект** в PyCharm по познатия начин и му дайте описателно име, например "while\_loop\_lab\_solutions"
  2. В проекта създайте нов Python файл с име "numbers\_in\_range"
  3. Прочетете входните данни от конзолата
- ```
number = int(input())
```
4. Направете **while** цикъл, който да се повтаря, **докато** съответното число **е по-малко от 1 или по-голямо от 100**:

```
while number < 1 or number > 100:
```

5. Ако програмата **влезе в тялото на цикъла**, това ще означава, че потребителят е въвел число, което не е в искания диапазон, затова първо отпечатайте съобщение "Invalid number!", след което **прочетете ново число** и **присвоете стойността му на променливата number**:

```
while number < 1 or number > 100:  
    print('Invalid number!')  
    number = int(input())
```

6. **След тялото на цикъла** отпечатайте променливата **number** в желанния формат. Тя ще има стойност **между 1 и 100**, понеже програмата ще е **излязла от цикъла**:

```

number = int(input())

while number < 1 or number > 100:
    print('Invalid number!')
    number = int(input())

print(f'The number is: {number}.')

```

7. Стартирайте програмата с **Ctrl+Shift+F10** и я тествайте с различни входни стойности.

## 2. Парола

Напишете програма, която първоначално прочита име и парола на потребителски профил. След това чете парола за вход, при въвеждане на грешна парола, потребителя да се подкани да въведе нова парола.

### Примерен вход и изход

| ВХОД                                  | ИЗХОД          |
|---------------------------------------|----------------|
| Nakov<br>1234<br>pass<br>1324<br>1234 | Welcome Nakov! |

| ВХОД                      | ИЗХОД          |
|---------------------------|----------------|
| Gosho<br>secret<br>secret | Welcome Gosho! |

### Насоки

1. Инициализирайте две променливи **username** и **password**, които ще съдържат потребителското име и паролата:

```

username = input()
password = input()

```

2. Инициализирайте променлива **data**, която ще държи въведената от потребителя парола за вход:

```

data = input()

```

3. В **while** цикъл, до въвеждане на валидна парола, четете нова:

```

while data != password:
    data = input()

```

4. Когато се въведе **валидна парола** принтирайте съобщението за успешен вход:

```

username = input()
password = input()

data = input()
while data != password:
    data = input()
print(f"Welcome: {username}!")

```

### 3. Редица числа $2k+1$

Напишете програма, която чете число  $n$ , въведено от потребителя, и отпечатва **всички числа**  $\leq n$  от **редицата**: 1, 3, 7, 15, 31, .... Всяко следващо число се изчислява като умножим **предишното** с 2 и добавим 1.

#### Примерен вход и изход

| вход | изход  | вход | изход       | вход | изход             | вход | изход                   |
|------|--------|------|-------------|------|-------------------|------|-------------------------|
| 3    | 1<br>3 | 8    | 1<br>3<br>7 | 17   | 1<br>3<br>7<br>15 | 31   | 1<br>3<br>7<br>15<br>31 |

#### Насоки

1. Прочетете от конзолата цяло число;
2. Създайте променлива с **първоначална стойност** 1, която ще използвате за брояч:
3. Създайте **while** цикъл, който се повтаря, докато **броячът е по-малък** от числото, прочетено от конзолата:

```
number = int(input())  
counter = 1  
  
while counter <= number:
```

4. При всяко повторение на цикъла **принтирайте стойността на брояча** и му **прибавяйте дадената стойност**:

```
number = int(input())  
counter = 1  
  
while counter <= number:  
    print(counter)  
    counter = 2 * counter + 1
```

### 4. Баланс по сметка

Напишете програма, която пресмята колко общо пари има в сметката, след като направите определен брой вноски. На първия ред ще получите **колко вноски трябва да се направят**. На всеки следващ ред ще получавате сумата, която трябва да внесете в сметката, **докато не се достигне броят вноски**. При всяка получена сума на конзолата трябва да се извежда съобщението **"Increase: "** + сумата и тя да се **прибавя в сметката**. Ако получите число **по-малко от 0** на конзолата трябва да се изведе **"Invalid operation!"** и **програмата да приключи**. Когато програмата приключи, трябва да се принтира **"Total: "** + общата сума в сметката, закръглена до втория знак след десетичната запетая.

#### Примерен вход и изход

| вход  | изход           | вход  | изход              |
|-------|-----------------|-------|--------------------|
| 3     | Increase: 5.51  | 5     | Increase: 120      |
| 5.51  | Increase: 69.42 | 120   | Increase: 45.55    |
| 69.42 | Increase: 100   | 45.55 | Invalid operation! |
| 100   | Total: 174.93   | -150  | Total: 165.55      |

## 5. Най-голямо число

Напишете програма, която чете **n** на брой цели числа ( $n > 0$ ), въведени от потребителя, и намира най-голямото измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа по едно на ред.

### Примерен вход и изход

| вход | изход | вход | изход | вход | изход | вход | изход | вход | изход |
|------|-------|------|-------|------|-------|------|-------|------|-------|
| 2    | 100   | 3    | 20    | 4    | 99    | 1    | 999   | 2    | -1    |
| 100  |       | -10  |       | 45   |       | 999  |       | -1   |       |
| 99   |       | 20   |       | -20  |       |      |       | -2   |       |
|      |       | -30  |       | 7    |       |      |       |      |       |
|      |       |      |       | 99   |       |      |       |      |       |

## 6. Най-малко число

Напишете програма, която чете **n** на брой цели числа ( $n > 0$ ), въведени от потребителя, и намира **най-малкото** измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа по едно на ред.

### Примерен вход и изход

| вход | изход | вход | изход | вход | изход | вход | изход | вход | изход |
|------|-------|------|-------|------|-------|------|-------|------|-------|
| 2    | 99    | 3    | -30   | 4    | -20   | 1    | 999   | 2    | -2    |
| 100  |       | -10  |       | 45   |       | 999  |       | -1   |       |
| 99   |       | 20   |       | -20  |       |      |       | -2   |       |
|      |       | -30  |       | 7    |       |      |       |      |       |
|      |       |      |       | 99   |       |      |       |      |       |

## 7. Завършване

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред - неговите годишни оценки. Ученикът преминава в следващия клас, **ако годишната му оценка е по-голяма или равна на 4.00**. Ако оценката му е под 4.00, той ще повтори класа. При успешно завършване на **12-ти** клас да се отпечата:

"{име на ученика} graduated. Average grade: {средната оценка от цялото обучение}"

Стойността трябва да бъде форматирана до втория знак след десетичната запетая.

### Примерен вход и изход

| Вход  | изход                                | вход | изход                              |
|-------|--------------------------------------|------|------------------------------------|
| Pesho | Pesho graduated. Average grade: 5.37 | Ani  | Ani graduated. Average grade: 5.45 |
| 4     |                                      | 5    |                                    |
| 5.5   |                                      | 5.32 |                                    |
| 6     |                                      | 6    |                                    |
| 5.43  |                                      | 5.43 |                                    |
| 4.5   |                                      | 5    |                                    |
| 6     |                                      | 6    |                                    |
| 5.55  |                                      | 5.5  |                                    |
| 5     |                                      | 4.55 |                                    |
| 6     |                                      | 5    |                                    |
| 6     |                                      | 6    |                                    |



|           |  |           |  |
|-----------|--|-----------|--|
| 5.43<br>5 |  | 5.56<br>6 |  |
|-----------|--|-----------|--|

## 8. Завършване - част 2

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред - неговите годишни оценки. Ученикът преминава в следващия клас, ако годишната му оценка е по-голяма или равна на **4.00**. Ако ученикът бъде скъсан повече от един път, то той бива изключен и програмата приключва, като се отпечата **името на ученика и в кой клас е изключен**.

При успешно завършване на **12-ти** клас да се отпечата :

"{име на ученика} graduated. Average grade: {средната оценка от цялото обучение}"

В случай, че ученикът е изключен от училище, да се отпечата:

"{име на ученика} has been excluded at {класа, в който е бил изключен} grade"

Стойността трябва да бъде форматирана до втория знак след десетичната запетая.

### Примерен вход и изход

| вход                                                                           | изход                                | вход                                              | изход                             |
|--------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------|-----------------------------------|
| Gosho<br>5<br>5.5<br>6<br>5.43<br>5.5<br>6<br>5.55<br>5<br>6<br>6<br>5.43<br>5 | Gosho graduated. Average grade: 5.53 | Mimi<br>5<br>6<br>5<br>6<br>5<br>6<br>6<br>2<br>3 | Mimi has been excluded at 8 grade |

## Примерна изпитна задача

### 9. \*Преместване

На осемнадесетия си рожден ден Хосе взел решение, че ще се изнесе да живее на квартира. Опаковал багажа си в **кашони** и намерил подходяща обява за апартамент под наем. Той започва да пренася своя багаж **на части**, защото не може наведнъж. Има ограничено **свободно пространство** в новото си жилище, където може да разположи вещите, така че мястото да бъде подходящо за живеене.

Напишете **програма**, която изчислява **свободния обем** от жилището на **Хосе**, който остава, след като **пренесе багажа си**.

Всеки кашон е с точни размери: **1m x 1m x 1m**.

### Вход

Потребителят въвежда следните данни на отделни редове:

1. Широчина на свободното пространство - цяло число;
2. Дължина на свободното пространство - цяло число;
3. Височина на свободното пространство - цяло число;
4. На следващите редове (до получаване на команда "Done") - брой кашони, които се пренасят в квартирата - цели числа

Програмата трябва да приключи прочитането на данни при команда "Done" или ако свободното място свърши.

## Изход

Да се отпечата на конзолата **един** от следните редове:

- Ако стигнете до командата "Done" и има още свободно място:  
"{брой свободни куб. метри} Cubic meters left."
- Ако свободното място свърши преди да е дошла команда "Done":  
"No more free space! You need {брой недостигащи куб. метри} Cubic meters more."

## Примерен вход и изход

| Вход                                         | Изход                                             | Обяснение                                                                                                                         |
|----------------------------------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 10<br>10<br>2<br>20<br>20<br>20<br>20<br>122 | No more free space! You need 2 Cubic meters more. | $10 * 10 * 2 = 200$ кубични метра.<br>$20 + 20 + 20 + 20 + 122 = 202$ кубични метра.<br>$200 - 202 = 2$ недостигащи кубични метра |
| 10<br>1<br>2<br>4<br>6<br>Done               | 10 Cubic meters left.                             | $10 * 1 * 2 = 20$ кубични метра.<br>$4 + 6 = 10$ кубични метра.<br>$20 - 10 = 10$ кубични метра.                                  |