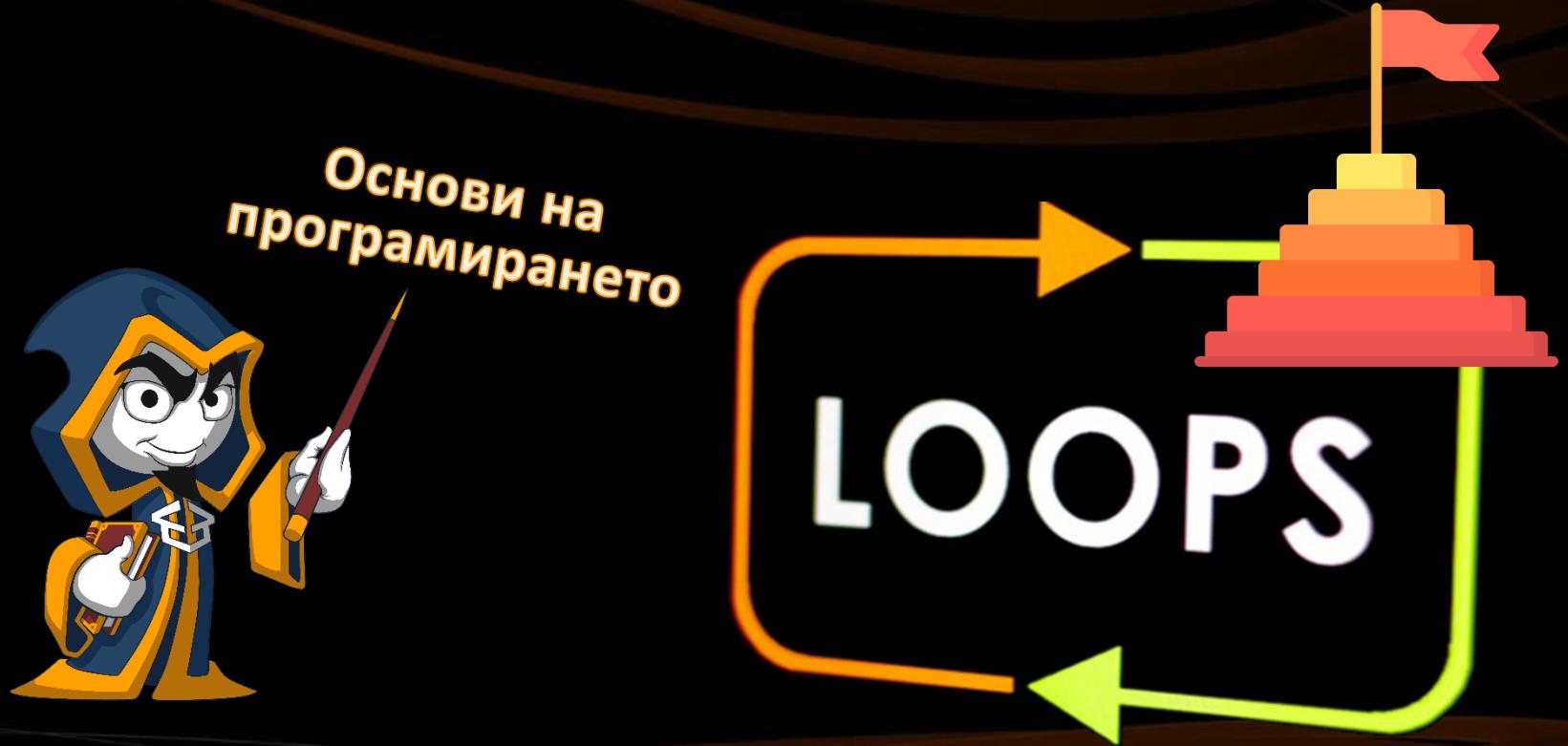


# Работа с по-сложни цикли

Цикли със стъпка, While-цикъл



Софтуни  
трейнърски екип  
Софтуерен университет  
<http://softuni.bg>



Имате въпроси?

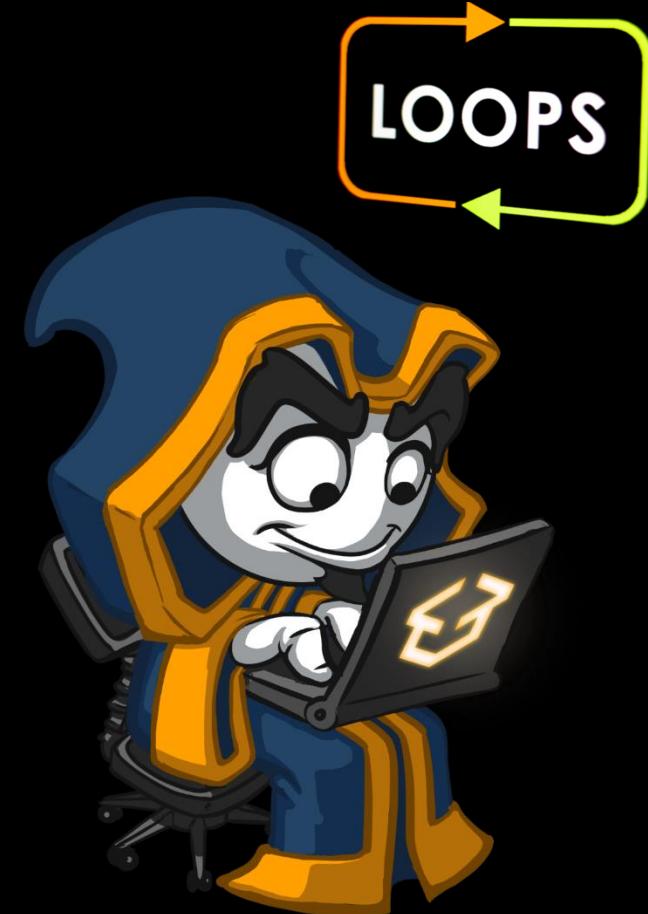


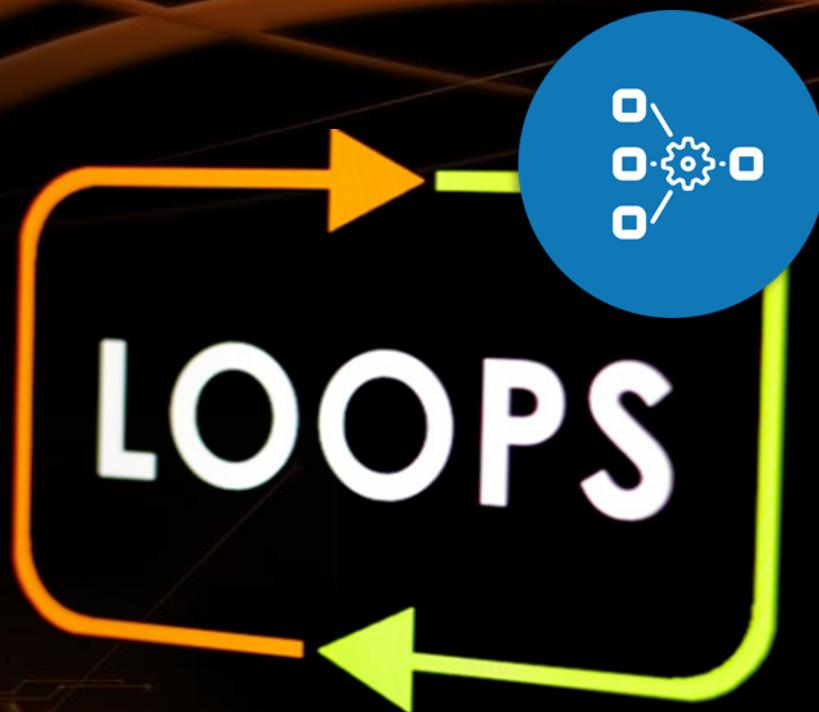
sli.do

#pb-april

# Съдържание

- По-сложни конструкции за цикъл:
  - For-цикъл със стъпка
  - For-цикъл с намаляваща стъпка
  - **while** цикъл
  - Безкраен цикъл и оператор **break**





## Цикли със стъпка

Работа с по-сложни `for`-цикли

# Числата от 1 до N през 3 - условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва числата от **1** до **n** със стъпка **3**
- Примерен вход и изход:

10 → 1, 4, 7, 10

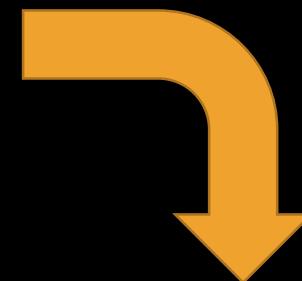
15 → 1, 4, 7, 10, 13



# Числата от 1 до N през 3 - решение

```
n = int(input())
for i in range(1, n+1, 3):
    print(i)
```

Задаване  
на стъпка



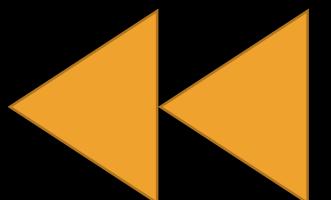
|   |          |
|---|----------|
|    | "C:\ ... |
|    | 15       |
|    | 1        |
|  | 4        |
|  | 7        |
|  | 10       |
|  | 13       |

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#0>

# Числата от N до 1 в обратен ред - условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва числата от **n** до **1** в обратен ред (**стъпка -1**)
- Примерен вход и изход:

100 → 100, 99, 98, ..., 3, 2, 1



BACKWARDS

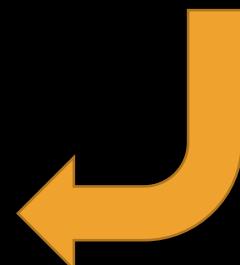
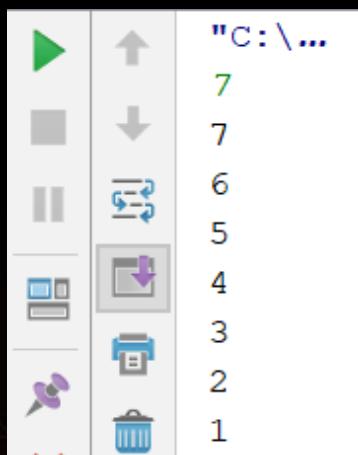
3 → 3, 2, 1

# Числата от N до 1 в обратен ред - решение

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

Намаляваща  
стъпка: -1

Обърнато условие:  
 $i \geq 1$



Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#1>

# Числата от 1 до $2^n$ с for-цикъл – условие

- Напишете програма, която:
  - Прочита цяло число  $n$
  - Отпечатва числата от 1 до  $2^n$
- Примерен вход и изход:

|    |     |   |   |
|----|-----|---|---|
| 8  | 2   | 8 |   |
| 16 | 128 | 4 | 2 |
| 2  | 4   | 2 | 4 |
| 2  | 32  | 8 |   |

10  1, 2, 4 , 8, 16, 32, ..., 1024

5  1, 2, 4, 8, 16, 32

# Числата от 1 до $2^n$ с for-цикъл - решение

```
n = int(input())
num = 1
for i in range(n+1):
    print(num)
    num = num * 2
```

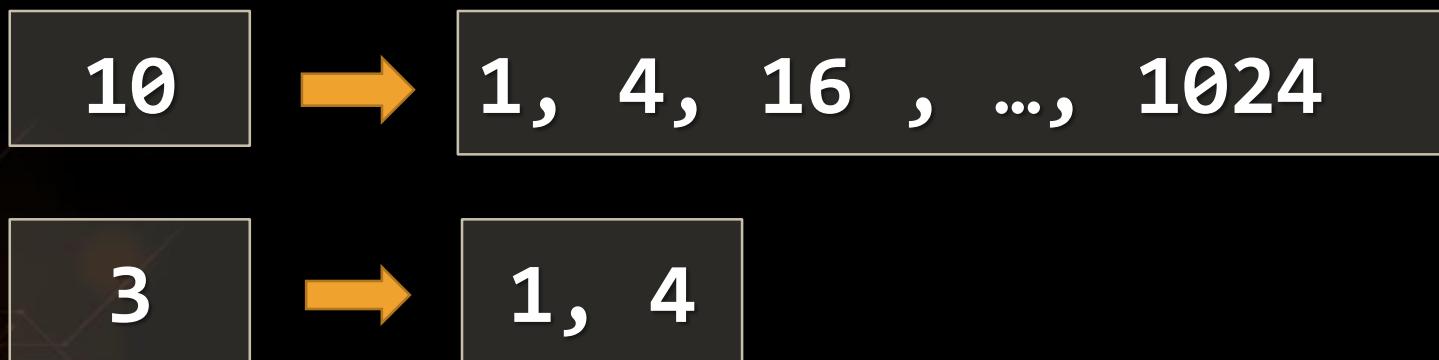


|  |  |          |
|--|--|----------|
|  |  | "C:\ ... |
|  |  | 5        |
|  |  | 1        |
|  |  | 2        |
|  |  | 4        |
|  |  | 8        |
|  |  | 16       |
|  |  | 32       |

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#2>

# Четни степени на 2 - условие

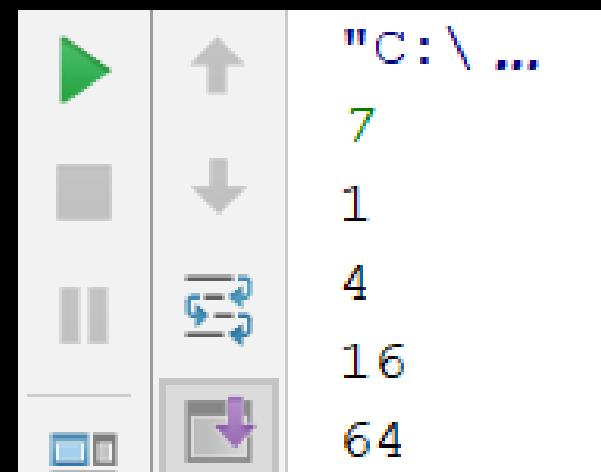
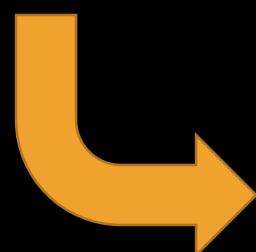
- Напишете програма, която:
  - Прочита цяло число  $n$
  - Отпечатва четните степени на  $2$  до  $2^n$ :  $2^0, 2^2, 2^4, 2^8, \dots, 2^n$
- Примерен вход и изход:



# Четни степени на 2 - решение

```
n = int(input())
num = 1
for i in range(0, n+1, 2):
    print(num)
    num = num * 2 * 2
```

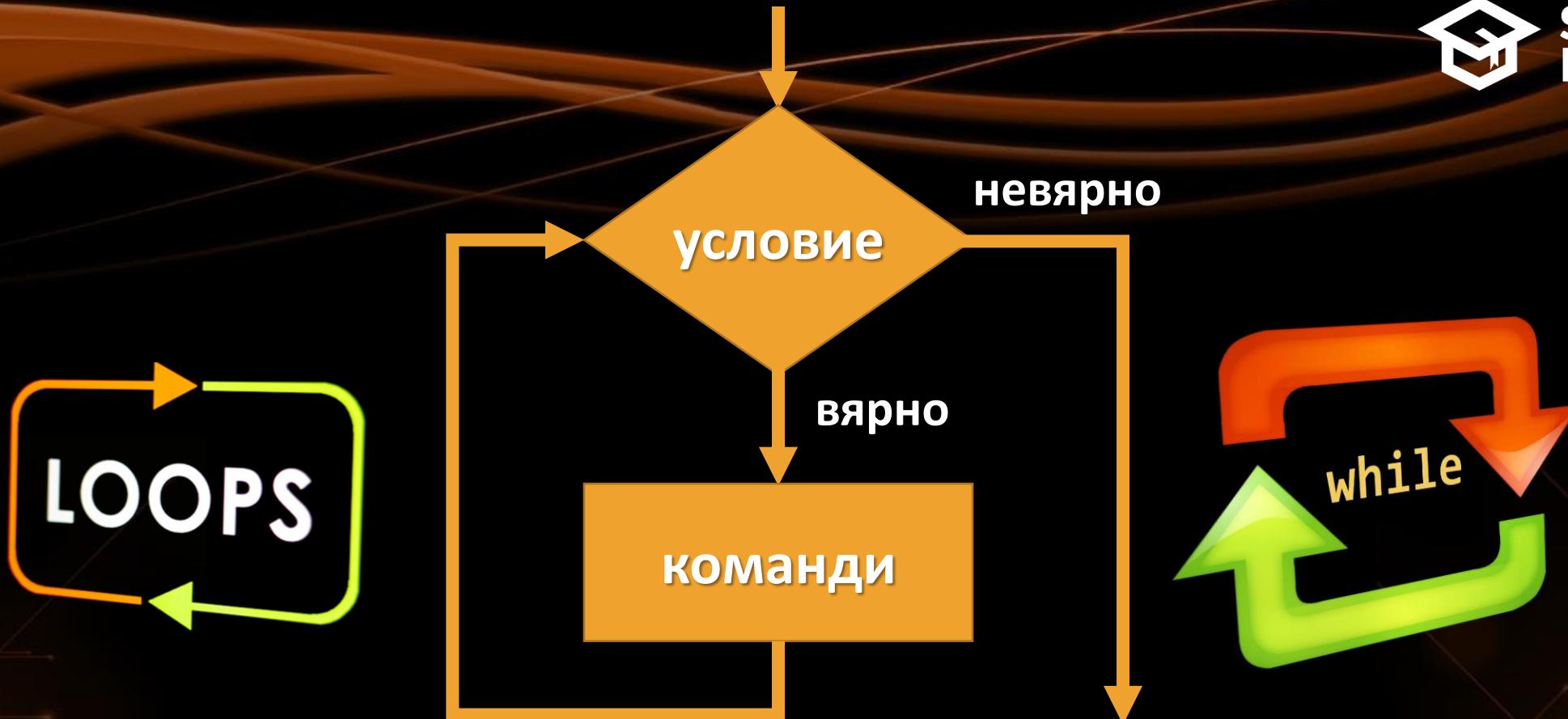
Ползваме  
стъпка 2



"C:\> ...

|    |
|----|
| 7  |
| 1  |
| 4  |
| 16 |
| 64 |

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#3>

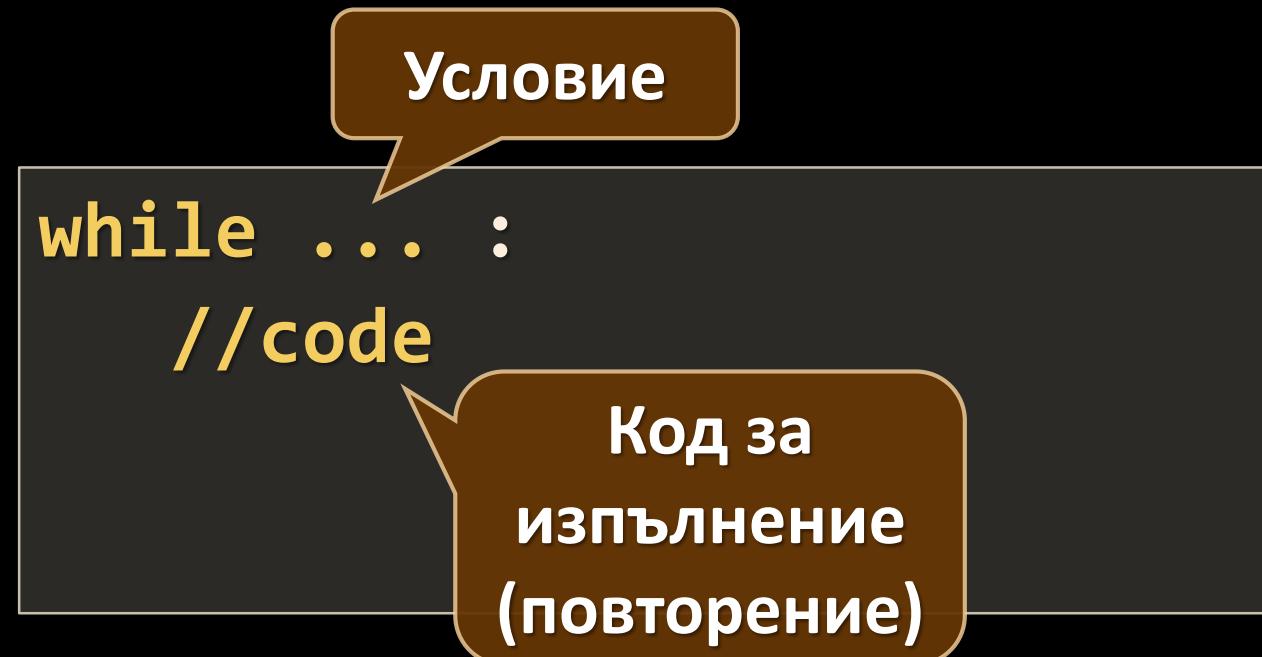
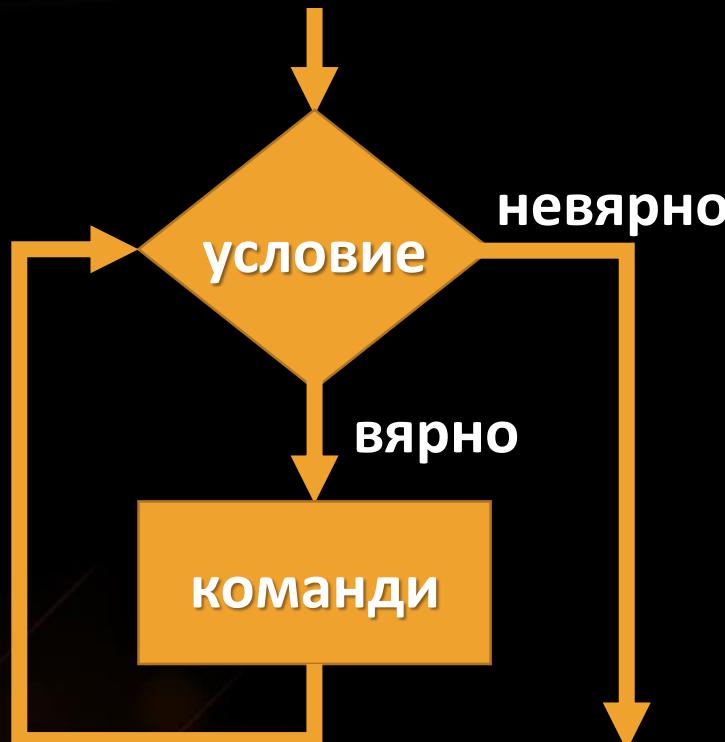


## While цикъл

Повторение докато е в сила дадено условие

# While цикъл

- Тялото на цикъла се изпълнява **докато е вярно** дадено условие



# Редица числа $2k+1$ - условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва всички числа  $\leq n$  от редицата: **1, 3, 7, 15, 31, ...**
  - Всяко следващо число е равно на предишното  $* 2 + 1$

**1,  $(1*2)+1 = 3$ ,  $(3*2)+1 = 7$ ,  $(7*2)+1 = 15 \dots$**

- Примерен вход и изход:

**17**



**1, 3, 7, 15**

**33**

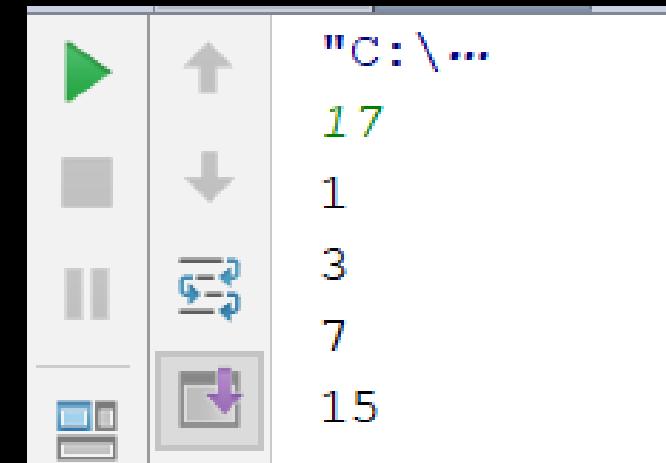


**1, 3, 7, 15, 31**

# Редица числа $2k+1$ - решение

```
n = int(input())
k = 1
while k <= n:
    print(k)
    k = 2 * k + 1
```

Повторение докато е  
в сила условието  $k \leq n$

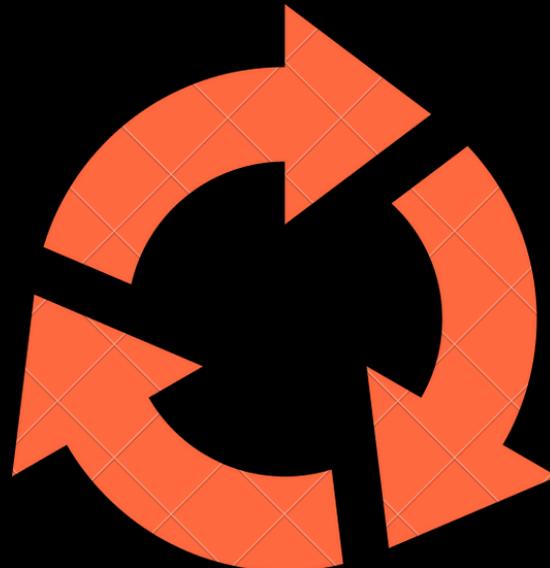


"C:\..."  
17  
1  
3  
7  
15

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#4>

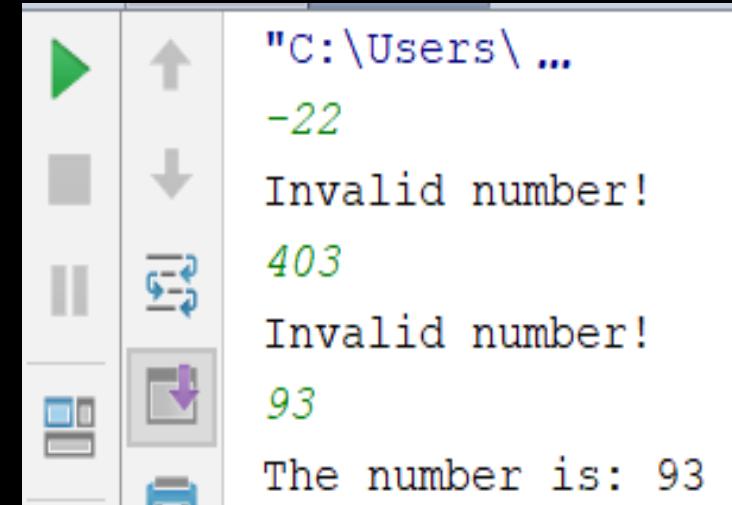
# Число в диапазона [1...100] - условие

- Напишете програма, която:
  - Прочита цяло число
  - Проверява дали е в диапазона [1...100]
  - При:
    - Намиране на число в диапазона,  
прекратява изпълнение
    - Невалидно число прочита ново



# Число в диапазона [1...100] - решение

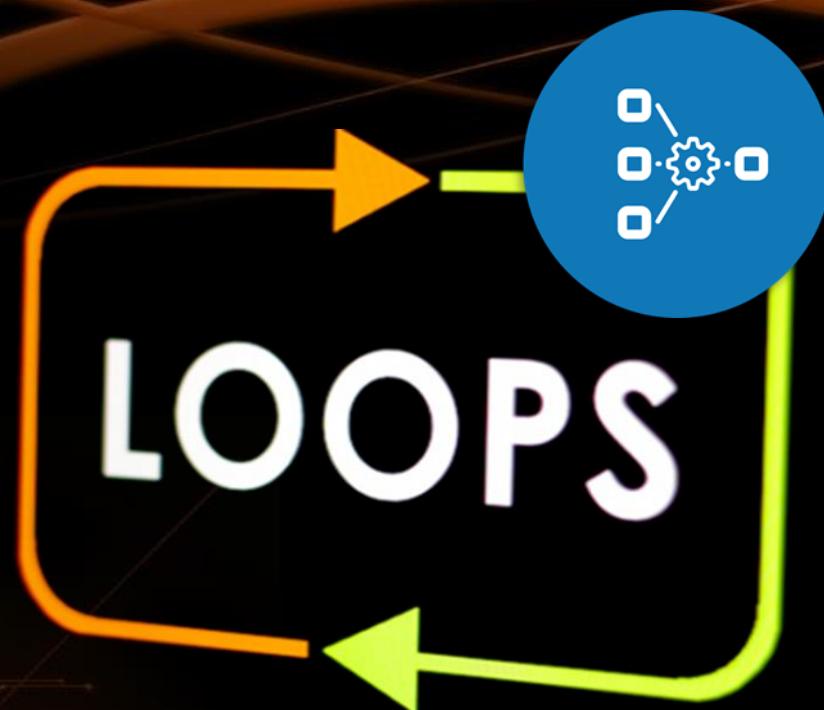
```
num = int(input())
while num < 1 or num > 100:
    print('Invalid number!')
    num = int(input())
print('The number is:', num)
```



|   |                   |
|---|-------------------|
| ▶ | "C:\Users\ ...    |
| ↑ | -22               |
| ↓ | Invalid number!   |
| ↔ | 403               |
| ↓ | Invalid number!   |
| ↑ | 93                |
| ▶ | The number is: 93 |

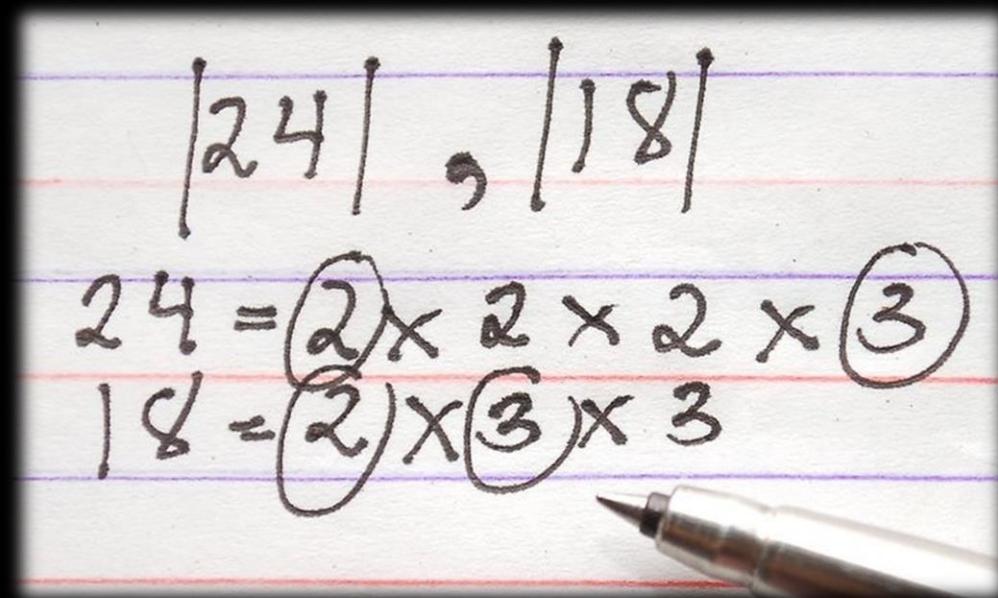


Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#5>



# Цикли със стъпка и while цикъл

Работа на живо в клас (лаб)

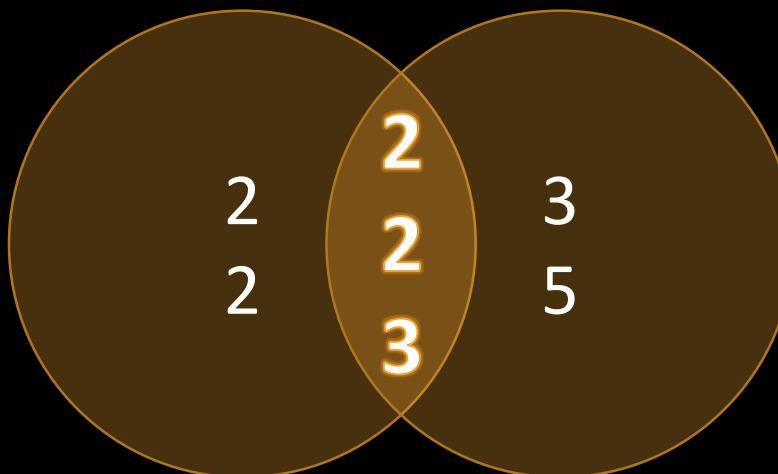


# Най-голям общ делител (НОД)

Алгоритъм на Евклид

# Най-голям общ делител (НОД)

- Най-голям общ делител (НОД) на две естествени числа **a** и **b** е най-голямото число, което дели едновременно **a** и **b** без остатък
  - $\text{НОД}(24, 16) = 8$
  - $\text{НОД}(67, 18) = 1$
  - $\text{НОД}(12, 24) = 12$
  - $\text{НОД}(15, 9) = 3$
  - $\text{НОД}(10, 10) = 10$
  - $\text{НОД}(100, 88) = 4$



# Алгоритъм на Евклид за НОД - условие

## Напишете програма, която:

- Прочита 2 цели числа **a** и **b**
- Намира най-големия им общ делител - **НОД(a, b)**

## Насоки:

- Докато не се достигне остатък 0:
  - Дели се по-голямото число на по-малкото
  - Взема се остатъка от делението

Псевдо код:

```
while b ≠ 0
    oldB = b;
    b = a % b;
    a = oldB;
print a;
```

# Алгоритъм на Евклид за НОД

- Да се въведат цели числа **a** и **b** и да се намери **НОД(a, b)**

```
a = int(input())
b = int(input())

while b != 0:
    oldB = b
    b = a % b
    a = oldB

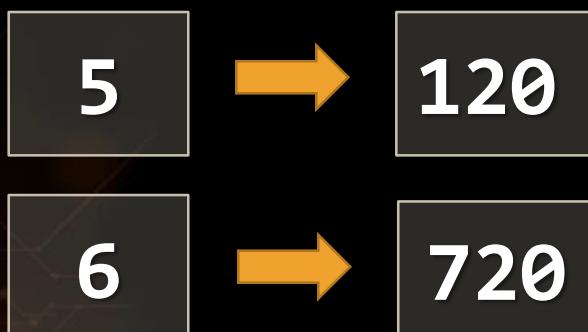
print('GCD =', a)
```

a = 600, b = 136  
600 / 136 = 4 (ост. 56)  
136 / 56 = 2 (ост. 24)  
56 / 24 = 2 (ост. 8)  
24 / 8 = 3 (ост. 0)  
НОД = 8

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#6>

# Изчисляване на факториел - условие

- Напишете програма, която:
  - Прочита естествено число **n**
  - Изчислява факториел от n ( $n!$ )
- Примерен вход и изход:
  - $5! = 1 * 2 * 3 * 4 * 5 = 120$



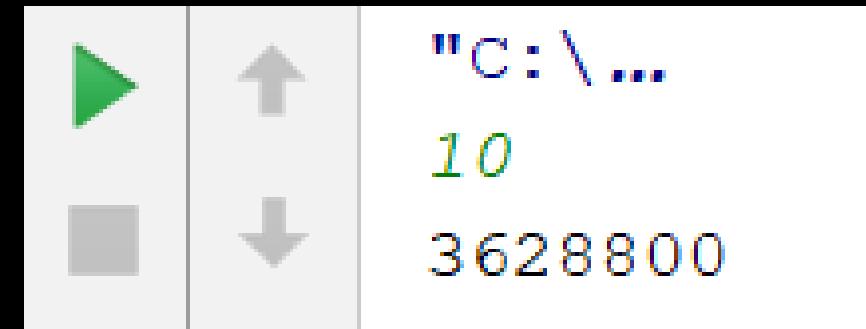
**$n!$**

# Изчисляване на факториел - решение

```
n = int(input())
fact = 1

while n > 1:
    fact = fact * n
    n -= 1

print(fact)
```



$n!$

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#7>

# Сумиране на цифрите на число - условие

- Напишете програма, която:
  - Прочита цяло положително число **n**
  - Сумира цифрите на **n**
- Примерен вход и изход:
  - $n = 5634: 5 + 6 + 3 + 4 = 18$

5634



18

1234



10



# Сумиране на цифрите на число - решение

```
n = int(input())
sum = 0
while n > 0:
    sum = sum + (n % 10)
    n = n // 10
print("Sum of digits: " + str(sum))
```

**n % 10** връща последната цифра на числото **n**

**n // 10** изтрива последната цифра на **n**

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#8>



# Безкрайни цикли и оператор break

- Безкраен цикъл – повтаряне на блок от код до безкрайност:

Условието е  
винаги вярно

```
while True:  
    print("Infinite loop")
```

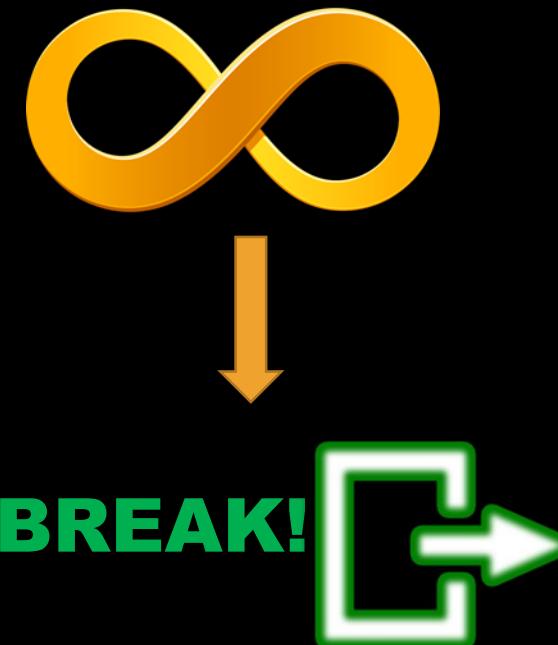


# Безкраен цикъл (2)

- Оператор **break** – прекъсване на цикъла

```
while True:  
    println("Infinite loop");  
    if ...  
        break
```

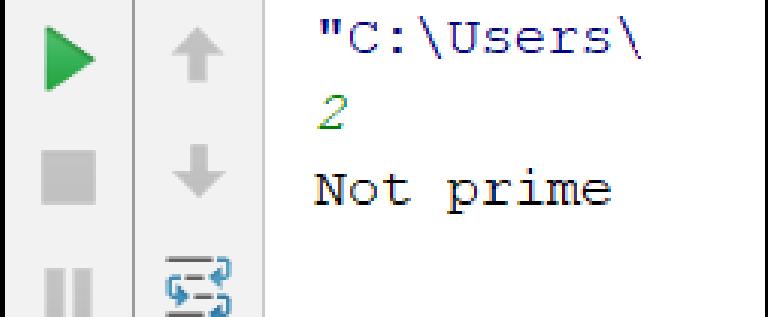
Условие за  
прекъсване на  
цикъла



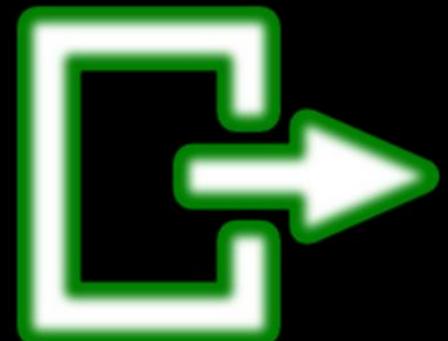
- Напишете програма, която:
  - Прочита цяло число **n**
  - Проверява да ли **n** е просто число
- Насоки:
  - Едно число **n** е **просто**, ако се дели единствено на **1** и **n**
  - Прости числа: **2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, ...**
  - Непрости (композитни) числа:  $10 = 2 * 5, 21 = 3 * 7, 143 = 13 * 11$

# Проверка за просто число - решение

```
n = int(input())
prime = True
i = 2
while i*i <= n:
    if n % i == 0:
        prime = False
        break
    i += 1
# Print if prime is found
```



```
"C:\Users\2
Not prime
```

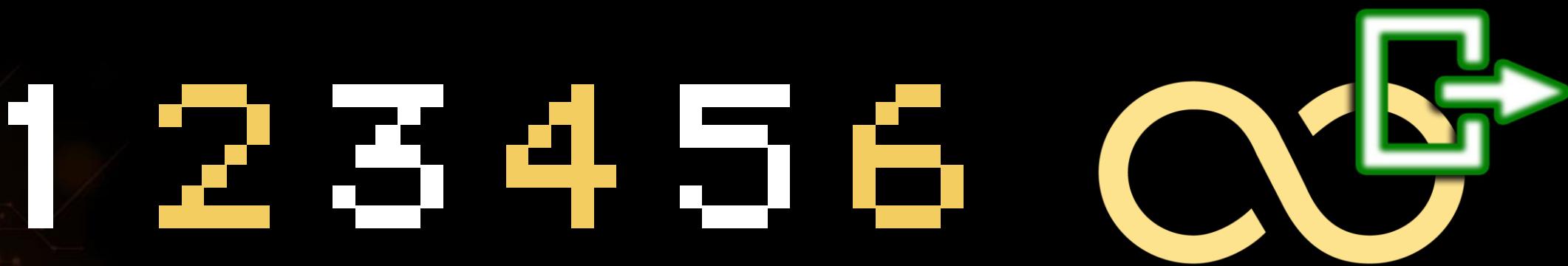


break излиза от  
текущия цикъл

# Print if prime is found

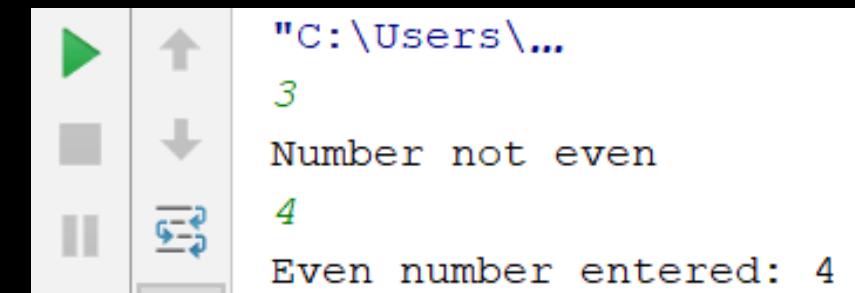
# Четно число - условие

- Напишете програма, която:
  - Прочита число **n**
  - Проверява дали **n** е четно
  - При невалидно число се връща към повторно въвеждане



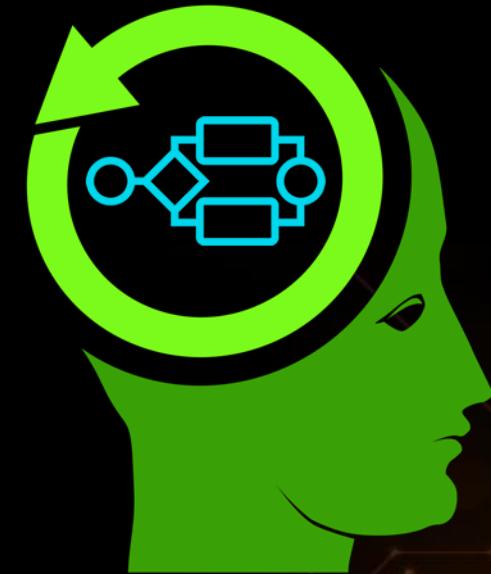
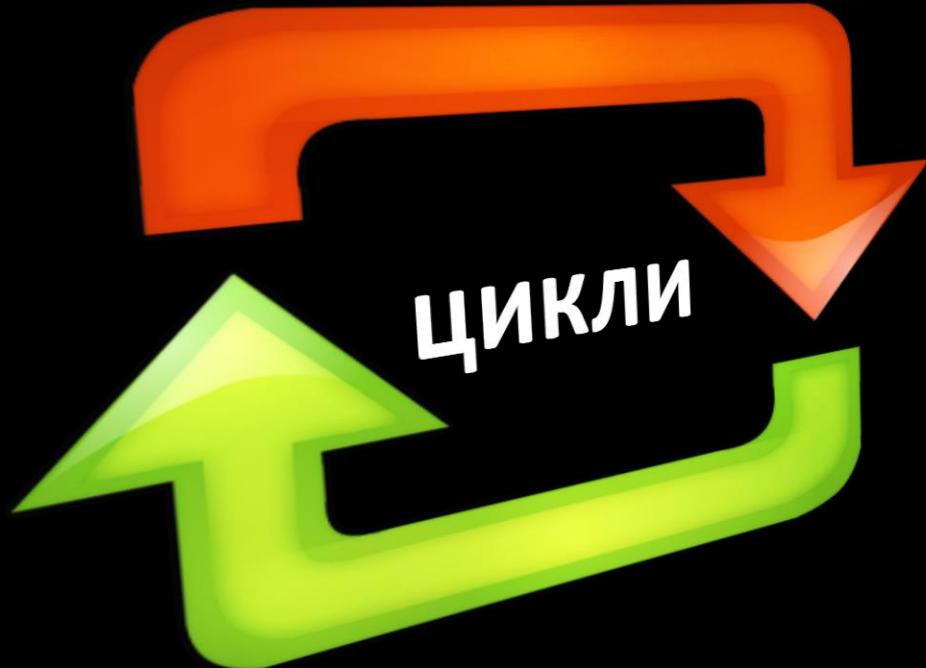
# Четно число - решение

```
while True:  
    n = int(input())  
    if n % 2 == 0:  
        break  
    print("Number not even")  
  
print("Even number entered: " + str(n))
```



```
"C:\Users\...\  
3  
Number not even  
4  
Even number entered: 4
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#10>



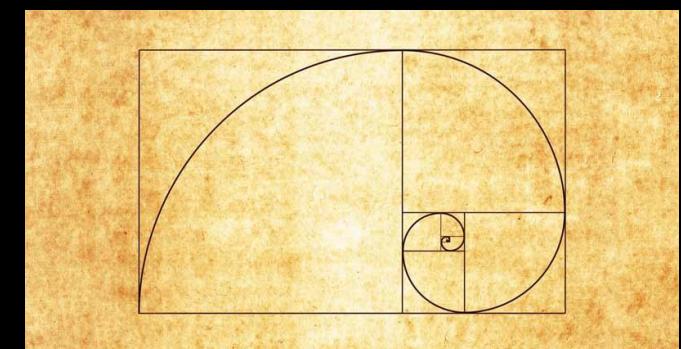
# Задачи с цикли

# Числа на Фибоначи - условие

- Напишете програма, която:
  - Прочита число **n**
  - Пресмята **n-тото** число на Фибоначи
- Числата на Фибоначи са следните: **1, 1, 2, 3, 5, 8, 13, 21, 34, ...**
- Примерен вход и изход:

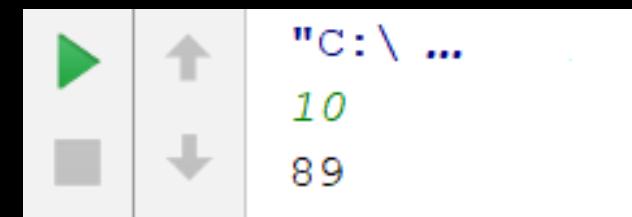
15 → 987

3 → 2



# Числа на Фиbonачи - решение

```
n = int(input())
f0 = 1
f1 = 1
for i in range(n-1):
    fNext = f0 + f1
    f0 = f1
    f1 = fNext
print(f1)
```



Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#11>

# Пирамида от числа – условие

- Да се отпечатат числата 1...n в пирамида като в примерите:

n = 7



|       |
|-------|
| 1     |
| 2 3   |
| 4 5 6 |
| 7     |

n = 10



|          |
|----------|
| 1        |
| 2 3      |
| 4 5 6    |
| 7 8 9 10 |

n = 12



|          |
|----------|
| 1        |
| 2 3      |
| 4 5 6    |
| 7 8 9 10 |
| 11 12    |

n = 15

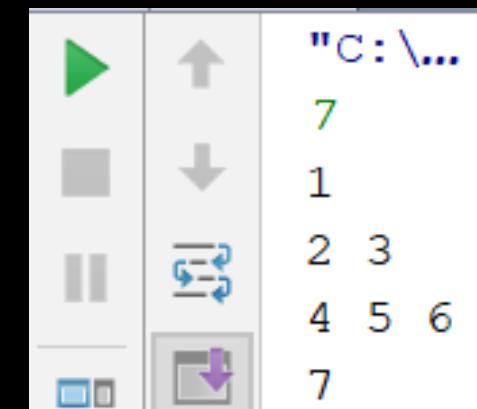


|                |
|----------------|
| 1              |
| 2 3            |
| 4 5 6          |
| 7 8 9 10       |
| 11 12 13 14 15 |



# Пирамида от числа – решение

```
n = int(input())
num = 1
for row in range(1, n+1):
    for col in range(row):
        if (col > 0):
            print(' ', end=' ')
        print(num, end=' ')
        num += 1
    if (num > n):
        break
    print()
if (num > n):
    break
```



|   |   |          |
|---|---|----------|
| ▶ | ↑ | "C:\..." |
| ■ | ↓ | 7        |
|   | ↔ | 1        |
|   |   | 2 3      |
|   |   | 4 5 6    |
|   | ▼ | 7        |

# Таблица с числа – условие

- Да се отпечатат числата 1...n в таблица като в примерите:

n = 2

↓

|   |   |
|---|---|
| 1 | 2 |
| 2 | 1 |

n = 3

↓

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |

n = 4

↓

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 3 |
| 3 | 4 | 3 | 2 |
| 4 | 3 | 2 | 1 |

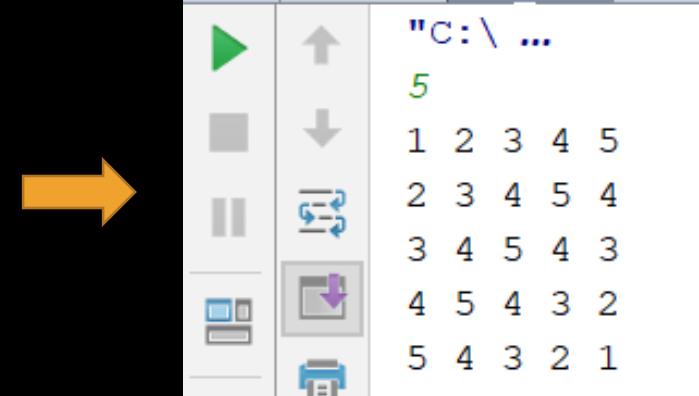
n = 5

↓

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 | 4 |
| 3 | 4 | 5 | 4 | 3 |
| 4 | 5 | 4 | 3 | 2 |
| 5 | 4 | 3 | 2 | 1 |

# Таблица с числа – решение

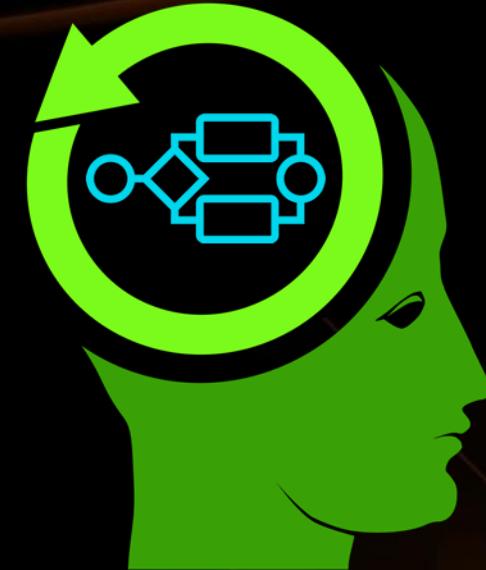
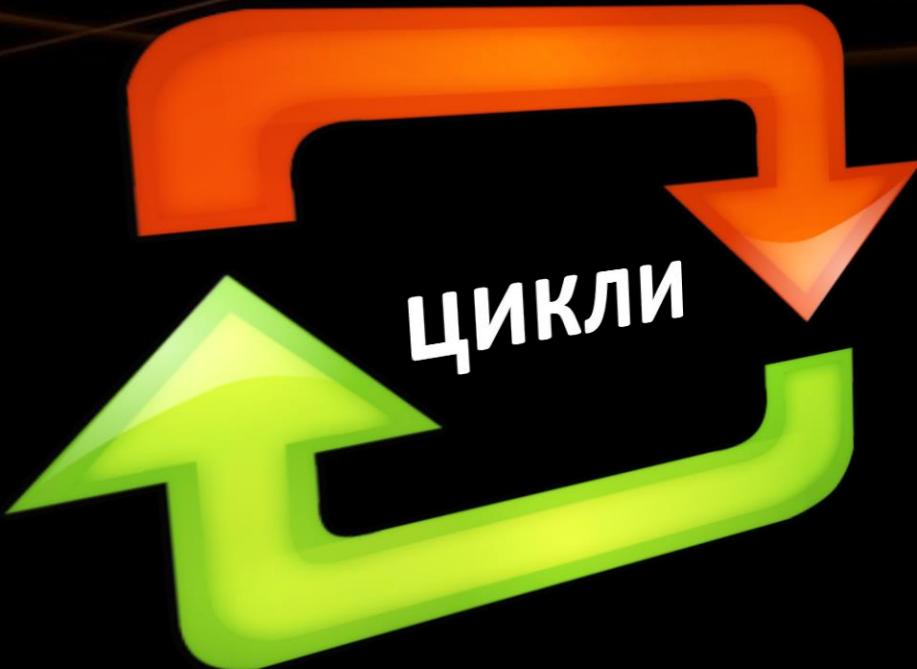
```
n = int(input())
for row in range(n):
    for col in range(n):
        num = row + col + 1
        if num > n:
            num = 2 * n - num
        print(str(num) + ' ', end=' ')
print()
```



|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

"C:\ ...  
5  
1 2 3 4 5  
2 3 4 5 4  
3 4 5 4 3  
4 5 4 3 2  
5 4 3 2 1

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/156#13>



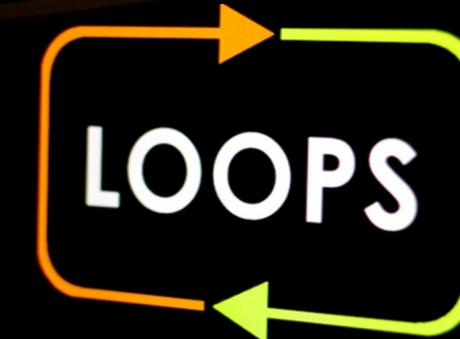
# По-сложни задачи с цикли

Работа на живо в клас (лаб)

# Какво научихме днес?

- Можем да ползваме **for**-цикли със стъпка:

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



- Цикли **while** повтаря докато е в сила  
данено условие:

```
num = 1  
  
while num <= n:  
    print(num)  
  
    num += 1
```



# Чертане с цикли



Questions?



# SoftUni Diamond Partners



**INDEAVR**  
Serving the high achievers

 **INFRASTRICS®**



**SoftwareGroup**  
*doing it right*



**NETPEAK**  
SEO and PPC for Business

**SUPER  
HOSTING**  
.**BG**

# SoftUni Diamond Partners



- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



- Благодарности: настоящият материал може да съдържа части от следните източници
  - Книга "Основи на програмирането със C#" от Светлин Наков и колектив с лиценз CC-BY-SA

# Бесплатни обучения в СофтУни



- Фондация "Софтуерен университет" – [softuni.org](http://softuni.org)
- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
  - [softuni.bg](http://softuni.bg)
- СофтУни @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)
- СофтУни форуми – [forum.softuni.bg](http://forum.softuni.bg)

