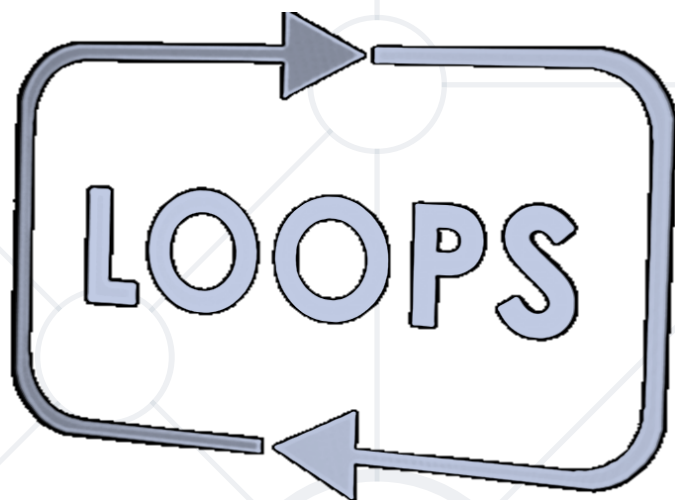


Повторения (цикли)

Прости повторения с for-цикъл



СофтУни

Преподавателски екип



Software
University



SoftUni
Foundation



Софтуерен университет
<http://softuni.bg>

Имате въпроси?

sli.do

#pb-jan

1. Преговор
2. Какво е цикъл?
3. for-цикъл – конструкция
4. Цикли със стъпка
5. Работа с текст
6. Техники за използване на for-цикли





Преговор

2. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
print(!(5 == 5) && (4 + 1 == 5));
```

True

False

Runtime
error

Compile time
error

3. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
print(!(3 == 3) || (3 == 5));
```

True

Runtime
error

False

Compile time
error

4. Какво ще се отпечата на конзолата, ако изпълним следната проверка:

```
print(!(3 > 5) || (1 == 1));
```

Compile time
error

Runtime
error

False

True

5. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
number = 101
if number >= 1:
    print("Larger than 1")
if number <= 101:
    print("Less than 101")
    print("Equal to 101")
```

Larger than 1

Less than 101

Equal to 101

Compile
time error

6. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
role = "Administrator"  
password = "SoftUni"  
if role == "SoftUni":  
    if password == "SoftUni":  
        print("Welcome!")
```

"Welcome!"

Runtime error

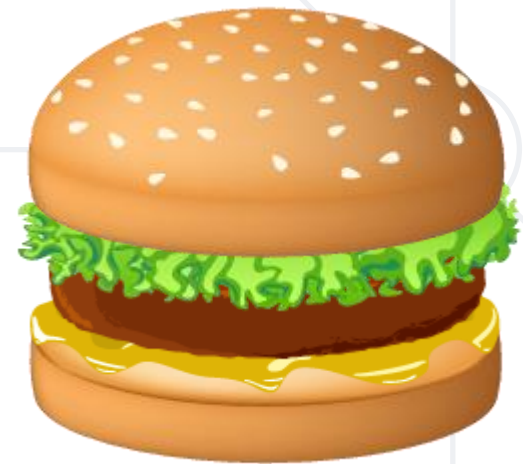
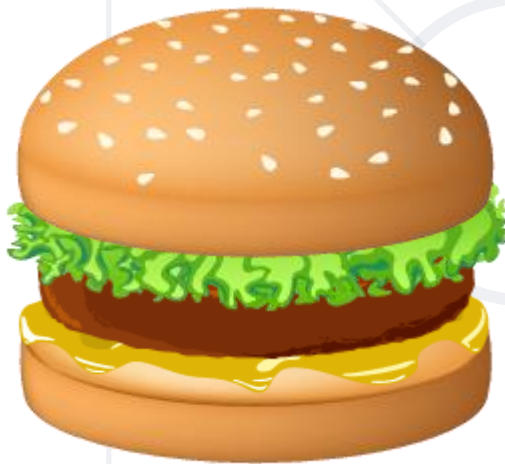
No output

Compile time
error



Цикли

- Често ни се налага да повтаряме едно и също действие многократно
- Ако искаме да направим 3 бургера, бихме повторили едни и същи действия 3 пъти:



Какво е цикъл? (2)

- Циклите в програмирането ни позволяват да повтаряме **едни и същи действия** определен брой пъти:

```
for i in range(1, 4):  
    print("Bottom bun")  
    print("Mustard")  
    print("Meat")  
    print("Lettuce")  
    print("Top bun")
```



Повторения на блокове код

Конструкция за for-цикъл

- Можем да повтаряме действия до определен момент чрез **for**-цикли

Ключова дума за
конструкцията

Начална
стойност

```
for i in range(1, 4):
```

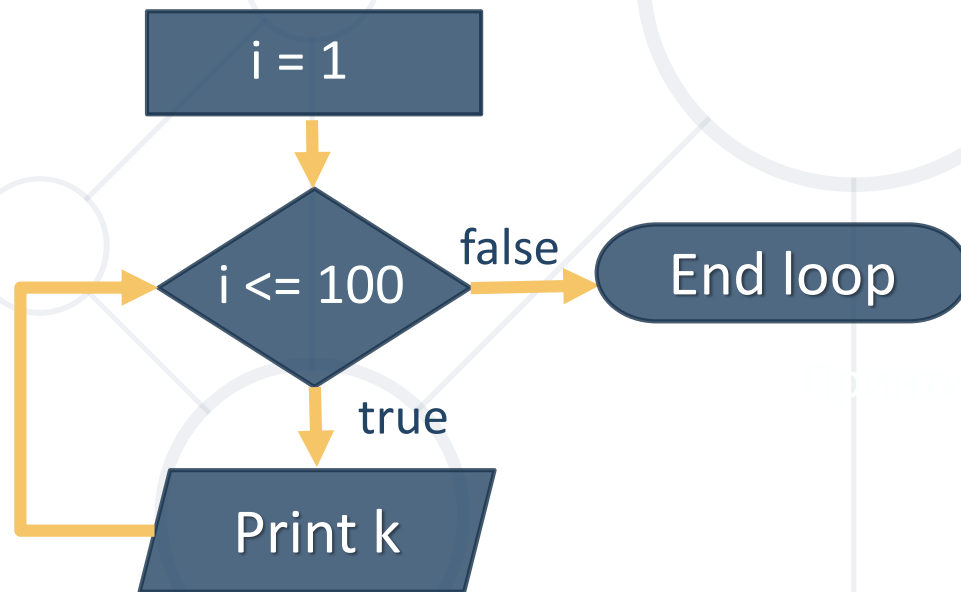
```
    print(i)
```

Крайна
стойност

Тяло на цикъла: блок от код за
повторение

Числа от 1 до 100

- Напишете програма, която:
 - Извежда числата [1...100] всяко на нов ред



Тестване на решението: <https://judge.softuni.bg/Contests/Compete/Index/1015#0>

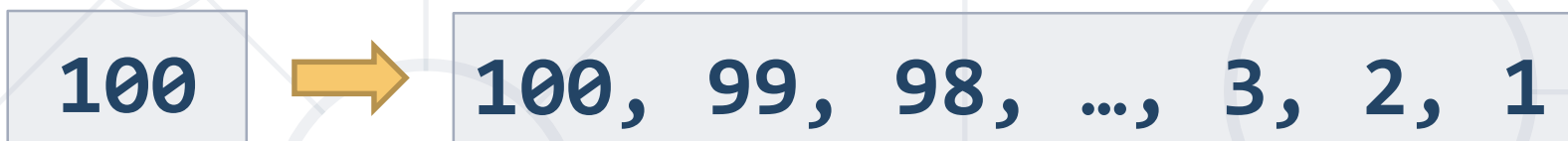


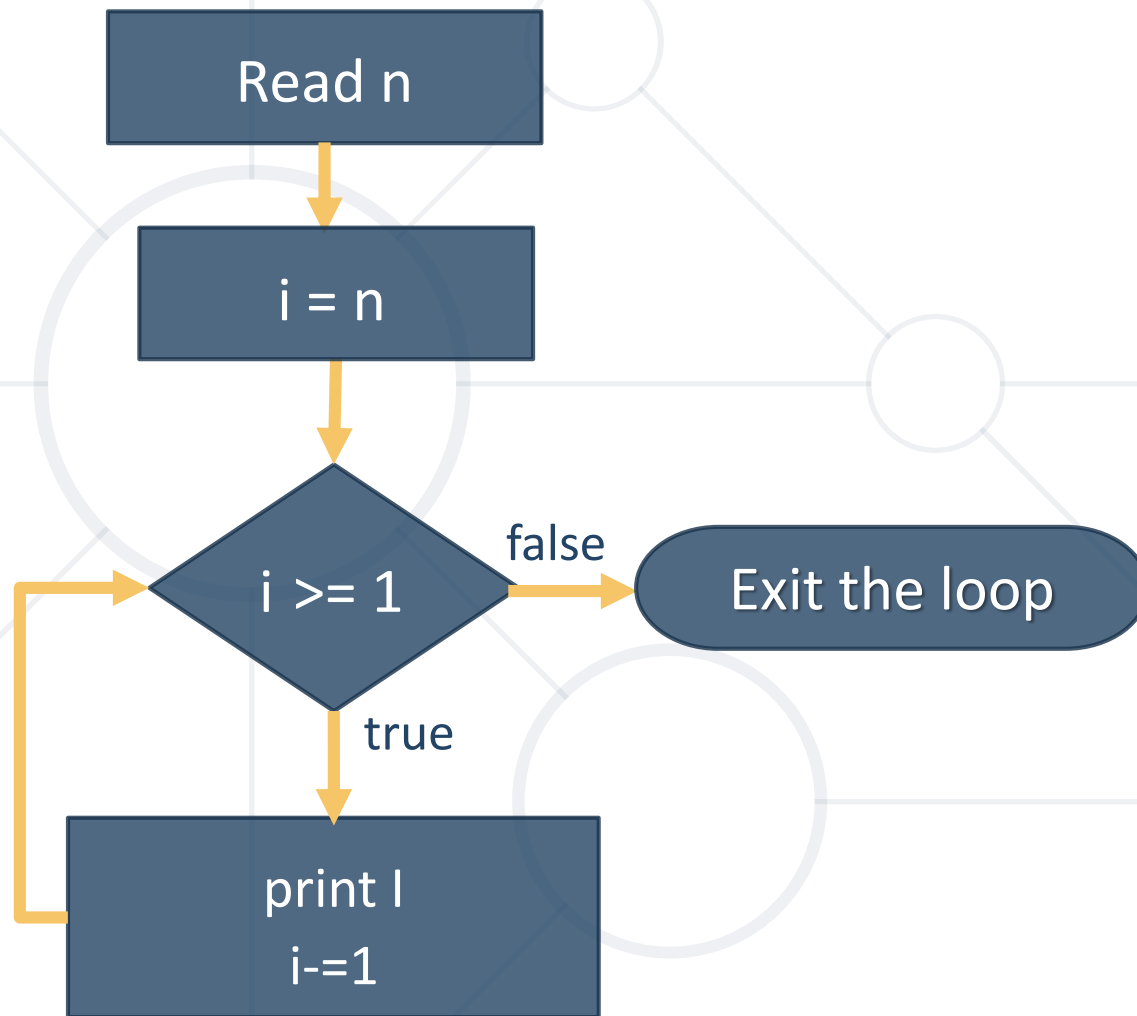
Работа с по-сложни for-цикли

Цикли със стъпка

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:





Числата от N до 1 в обратен ред – решение

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

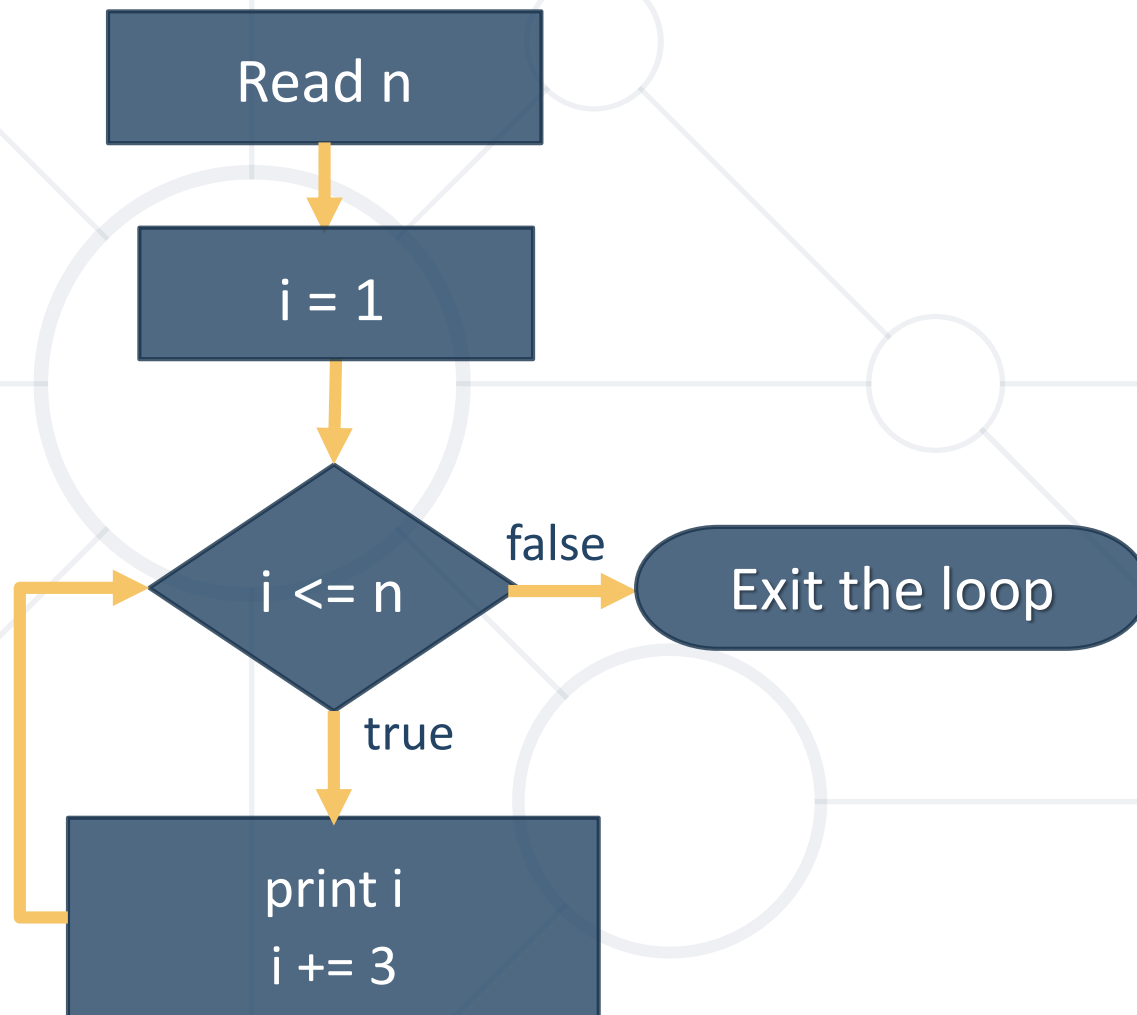
Обърнато условие: $i \geq 1$

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/1015#8>

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:





Числата от 1 до N през 3 – решение

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

Задаване на
стъпка 3

Четни степени на 2 – условие

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

10 → 1, 4, 16 , ..., 1024

7 → 1, 4, 16 , ..., 64

Четни степени на 2 – решение

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

Ползваме
стъпка 2



Работа с текст

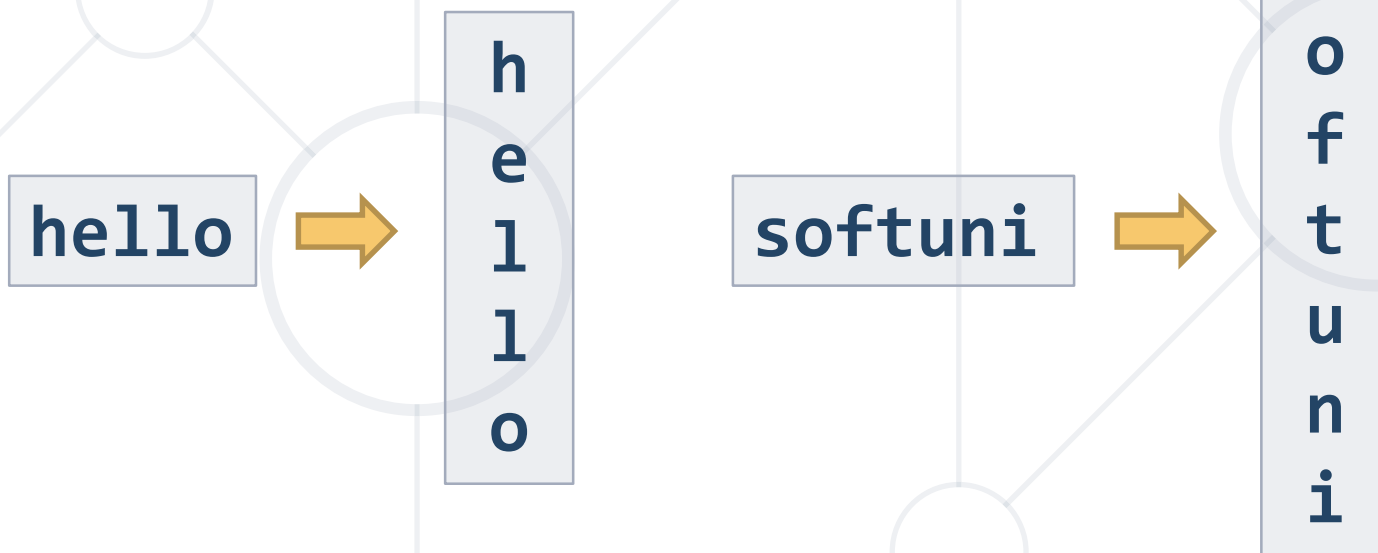
- Можем да вземем дължината на текст

```
text = input()      # въвеждаме SoftUni
length = len(text)  # 7
```

- Можем да вземем символ от текст по индекс

```
text = input()      # въвеждаме SoftUni
letter = text[4]     # U
```

- Напишете програма, която
 - чете текст(стринг)
 - печата всеки символ от текста на отделен ред
- Примерен вход и изход:



```
length = input()
for i in range(0, len(length)):
    print(length[i])
```

Взимаме дължината
на текста

Взимаме всеки
символ по индекс i

Сумиране на гласни букви - условие

- Напишете програма, която:
 - Прочита от потребителя текст
 - Извежда сумата на гласните букви според таблицата по-долу:

а	е	и	о	и
1	2	3	4	5

- Примерен вход и изход:

hello



6

(e+o = 2+4 = 6)

hi



3

(i = 3)

bamboo



9

(a+o+o = 1+4+4 = 9)

beer



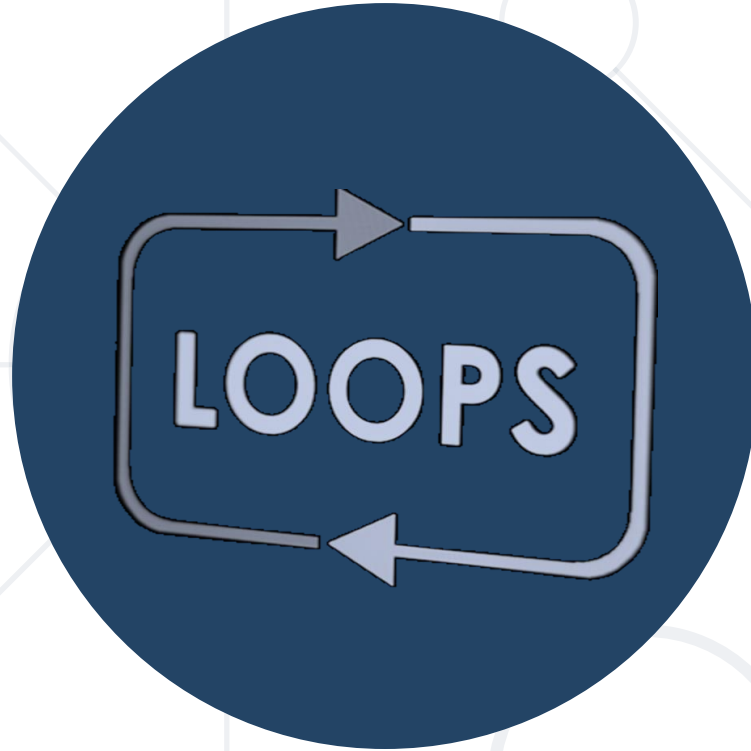
4

(e+e = 2+2 = 4)

Сумиране на гласни букви - решение

```
length = input()
sum = 0

for i in range(0, len(length)):
    if length[i] == "a":
        sum += 1
    if length[i] == "e":
        sum += 2
#TODO: make if statement for letters 'i' 'o' 'u'
print(f"Vowels sum = {sum}")
```



Повторения на блокове код

Решаване на задачи в клас (лаб)



Техники за използване на for-цикли

Задачи с цикли

- Напишете програма, която:
 - Прочита цяло число **n** от потребителя
 - Прочита **n** последователни пъти числа и ги сумира
 - Извежда пресметнатата сума
- Примерен вход и изход:

2
10
20



30

3
-10
-20
-30

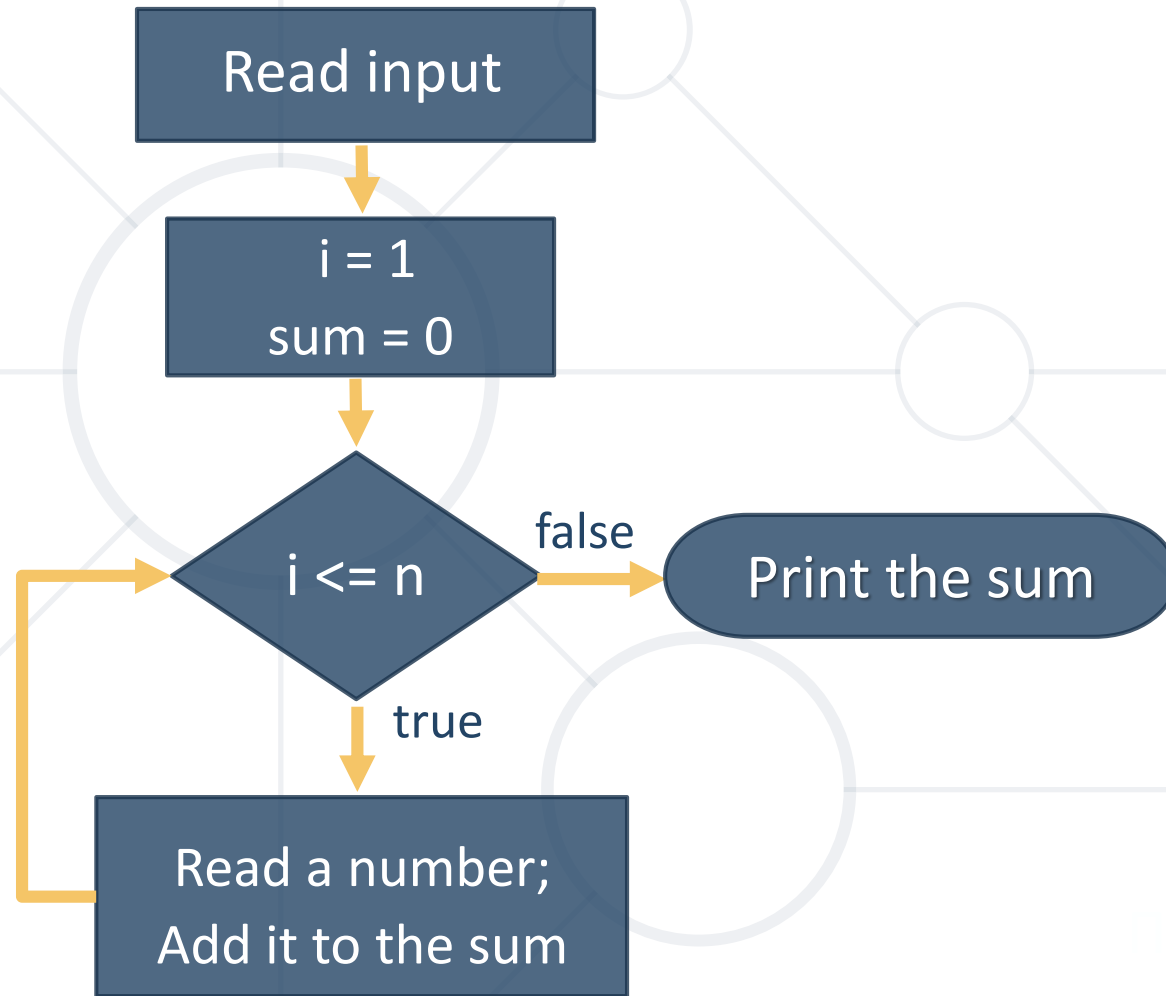


-60

4
45
-20
7
11



43



Редица цели числа - условие

- Напишете програма, която:
 - Чете **n** на брой цели числа
 - Принтира най-голямото и най-малкото число

5
10
20
304
0
50

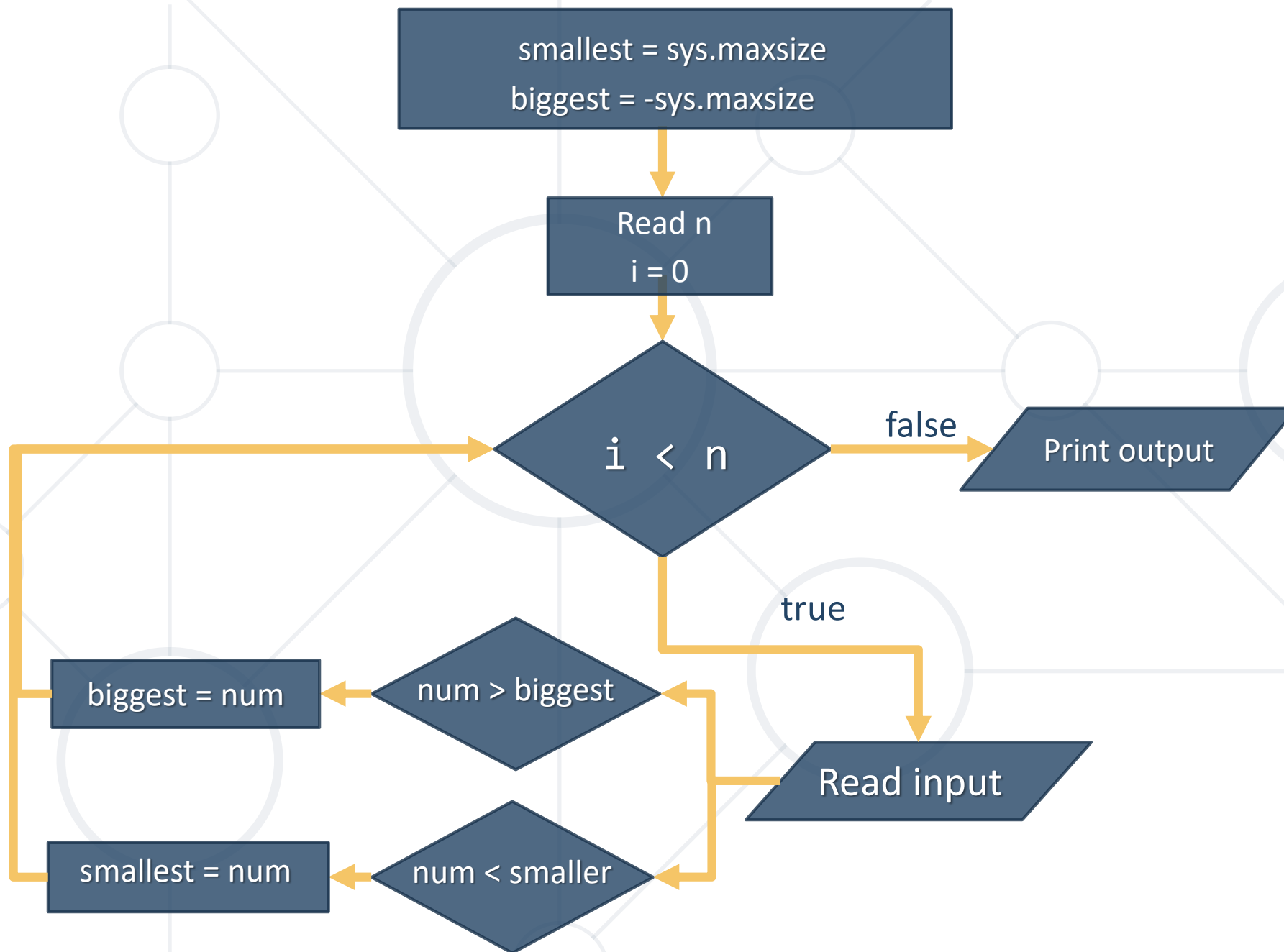


Max number: 304
Min number: 0

15
5
25
255
154
3



Max number: 255
Min number: 3



```
import sys
smallest = sys.maxsize
biggest = -sys.maxsize
n = int(input())
for i in range(0, n):
    num = int(input())
    if num < smallest:
        smallest = num
    if num > biggest:
        biggest = num

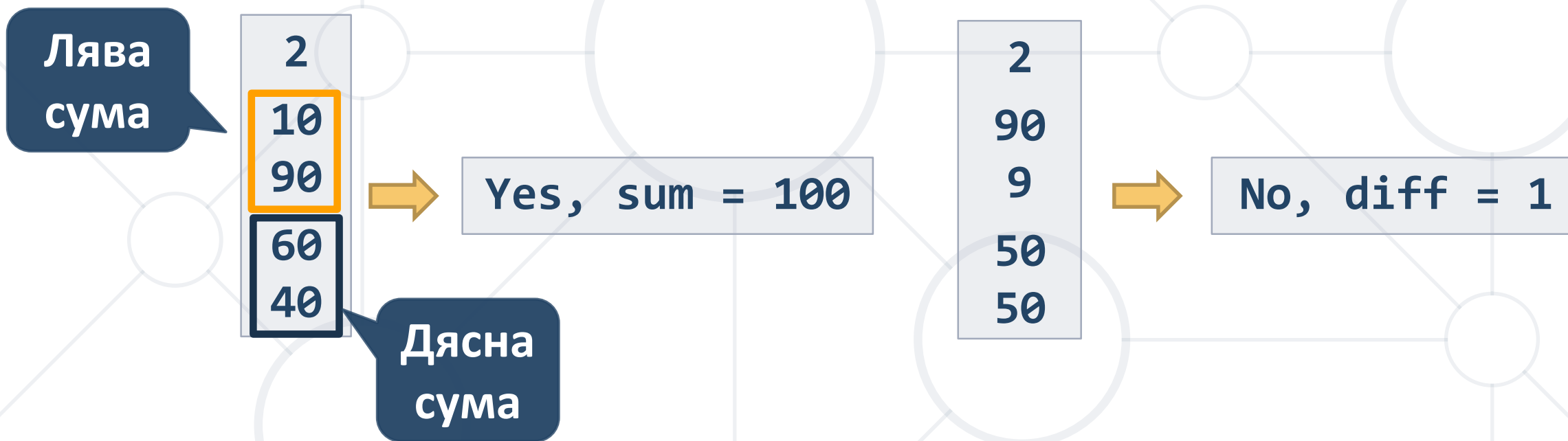
print(f"Max number: {biggest}")
print(f"Min number: {smallest}")
```

1 3 7

- Напишете програма, която:
 - Прочита цяло число n от потребителя
 - Прочита последователно $2*n$ числа
 - Проверява дали сумите на **левите** n и **десните** n числа са равни
 - При равенство извежда "Yes" и сумата, в противен случай - "No" и разликата (изчислена като положително число)

Лява и дясна сума - условие

- Примерен вход и изход:



Решение: лява и дясна сума

```
int(input())
leftSum = 0
for i in range(1, n + 1):
    leftSum = leftSum + int(input())

#TODO: read and calculate the rightSum
if leftSum == rightSum:
    print(f"Yes, sum = {leftSum}")
else:
    diff = abs(rightSum - leftSum)
    print(f"No, diff = {diff}")
```


- Напишете програма, която:
 - Прочита цяло число(n) от потребителя
 - Прочита последователно n на брой числа
 - Проверява дали сумата на числата на четни позиции е равна на сумата на числата на нечетни позиции
 - При равенство печата **"Yes"** и сумата; иначе печата **"No"** и разликата (положително число).

Четна / нечетна сума - условие

- Примерен вход и изход:

4
10
50
60
20



Yes
Sum = 70

4
3
5
1
-2



No
Diff = 1

3
5
8
1



No
Diff = 2

Решение: четна / нечетна сума

```
n = int(input())
oddSum = 0
evenSum = 0
for i in range(1, n + 1):
    element = int(input())
    if i % 2 == 0:
        evenSum += element
    else:
        oddSum += element
#TODO: print the sum / difference
```



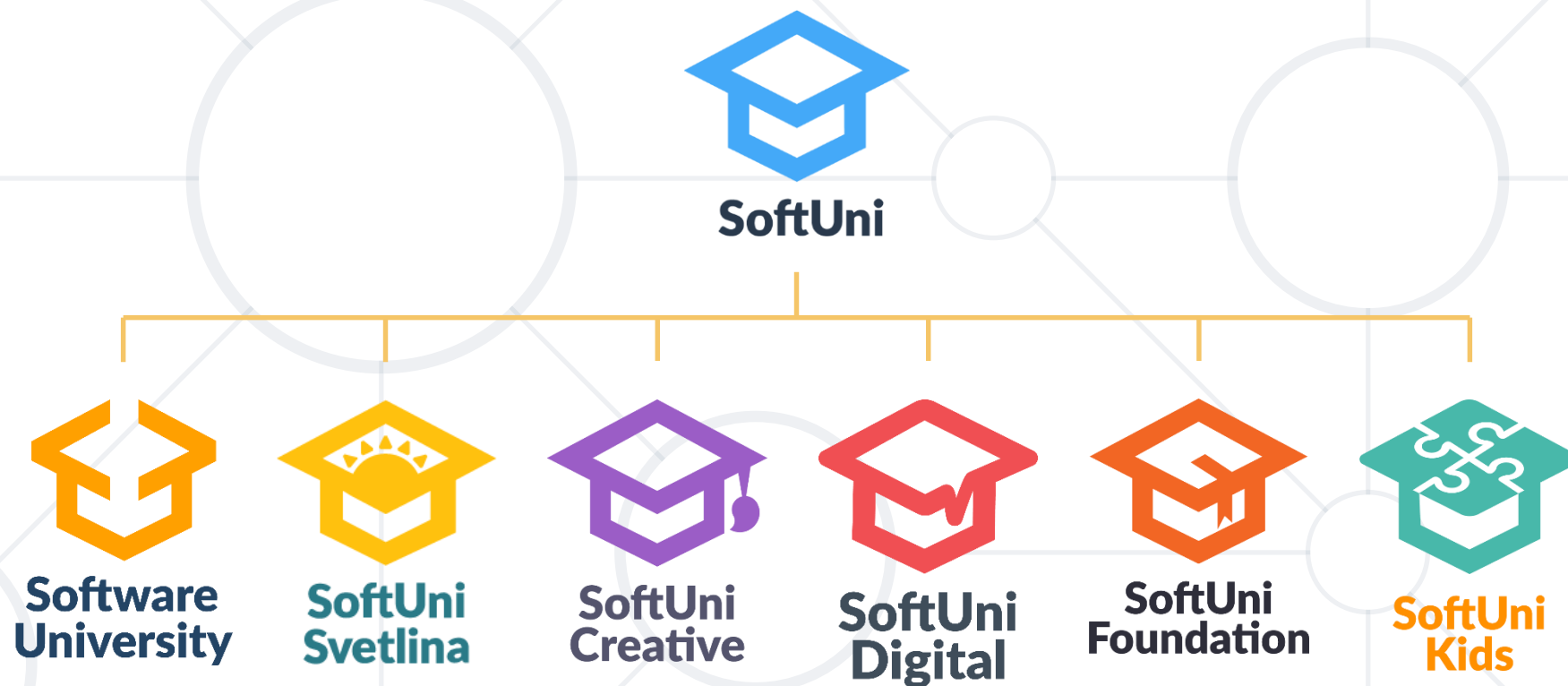
По-сложни задачи с цикли

Решаване на задачи в клас (лаб)

- Можем да повтаряме блок код с **for**-цикъл
- Цикли със стъпка
 - Цикли с намаляваща стъпка
- Можем да вземем символ по индекс от текст



Въпроси?



SoftUni Diamond Partners



XSsoftware



SBTech
we know sports



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре



INDEAVR

Serving the high achievers



INFRAGISTICS®



STEMO®
Computer Systems & Software

SUPERHOSTING.BG

SoftUni Organizational 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

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

