



Преговор

1. Колко пъти ще се изпише "SoftUni" на конзолата след изпълнението на следния код:

```
let i = 0;  
while(i <= 5){  
    console.log("SoftUni");  
    i++;  
}
```

5

0

4

6

1. Колко пъти ще се изпише "SoftUni" на конзолата след изпълнението на следния код:

```
let i = 0;  
while(i <= 5){  
    console.log("SoftUni");  
    i++;  
}
```

5

0

4

6

2. Колко пъти ще се изпише "SoftUni" на конзолата след изпълнението на следния код:

```
let i = 0;
while(i == 0){
  console.log("SoftUni");
  if(i == 1)
    break;
}
```

0

1

100000

Безброй
много пъти

2. Колко пъти ще се изпише "SoftUni" на конзолата след изпълнението на следния код:

```
let i = 0;
while(i == 0){
  console.log("SoftUni");
  if(i == 1)
    break;
}
```

0

1

100000

Безброй
много пъти

3. Какъв ще е резултатът от изпълнението на следния код:

```
let i = 0;  
while(i <= 10){  
  console.log("SoftUni");  
  i--;  
}
```

10

0

Безкраен
цикъл

11

3. Какъв ще е резултатът от изпълнението на следния код:

```
let i = 0;  
while(i <= 10){  
  console.log("SoftUni");  
  i--;  
}
```

10

0

Безкраен
цикъл

11

4. Какъв ще е резултатът от изпълнението на следния код:

```
let i = 0;
while (i < 6){
  i++;
  if (i % 2 == 0)
    console.log(i);
}
```

024

24

246

123456

4. Какъв ще е резултатът от изпълнението на следния код:

```
let i = 0;
while (i < 6){
  i++;
  if (i % 2 == 0)
    console.log(i);
}
```

024

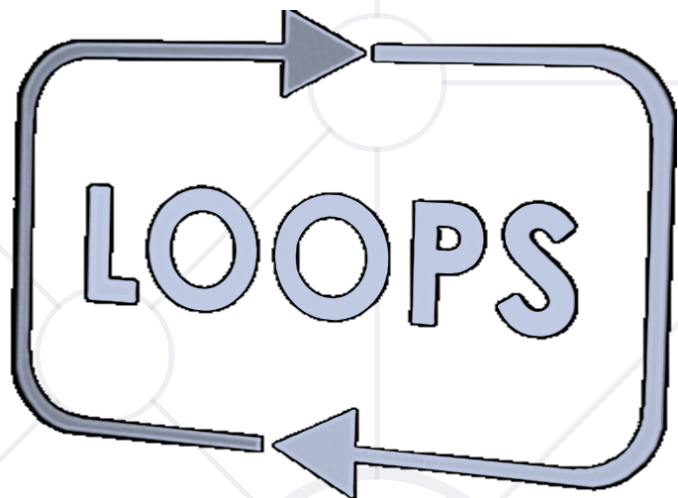
24

246

123456

Повторения (цикли)

Прости повторения с for-цикъл



СофтУни

Преподавателски екип



Software
University



SoftUni
Foundation



Софтуерен университет
<http://softuni.bg>

Имате въпроси?

sli.do

#pb-nov



Повторения на блокове код

Конструкция за for-цикъл

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

Ключова дума за
конструкцията

Начална
стойност

Крайна
стойност

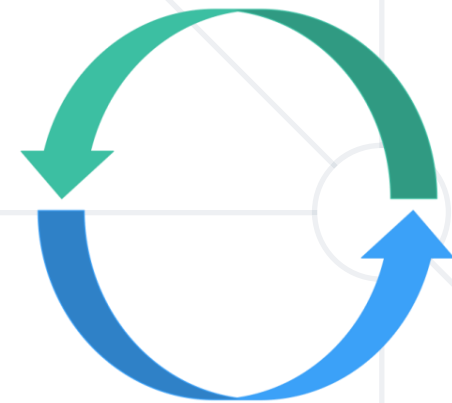
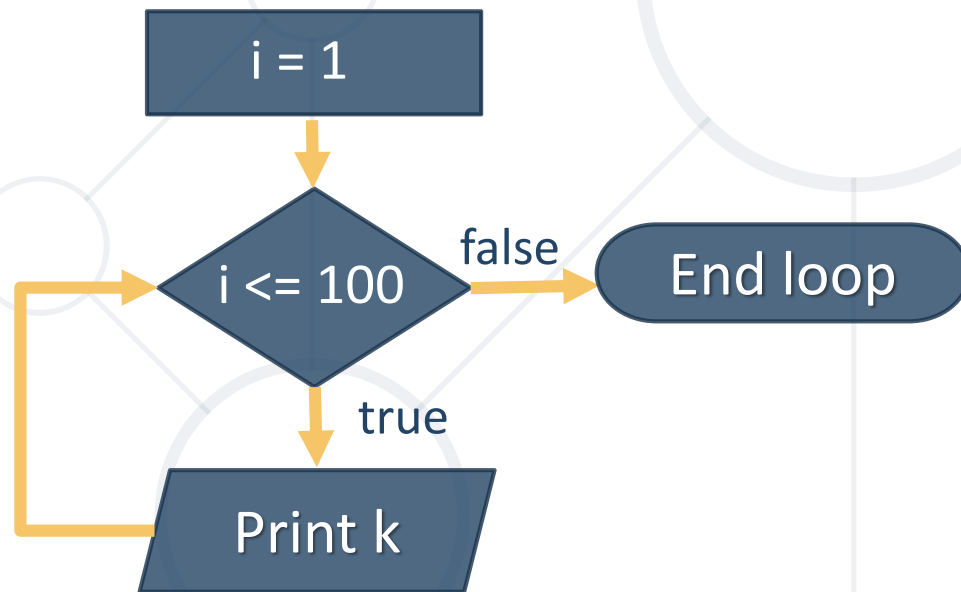
```
for (let i = 1; i <= 10; i++){  
  console.log(i);  
}
```

Инкрементация
на индекса (i)

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>

- Символите, които използваме се представят като числа
 - Поместени са в ASCII таблицата
- Примери (знак и неговата ASCII стойност):

а → 97

@ → 64

A → 65

+ → 43

- Съвременен вариант на ASCII таблицата
 - Съдържа 137 439 знака и обхваща 146 писмености
- Примери (знак и неговата Unicode стойност):

'\u005A' → Z

'\u007E' → ~

'\u00A2' → ¢

'\u00B5' → μ

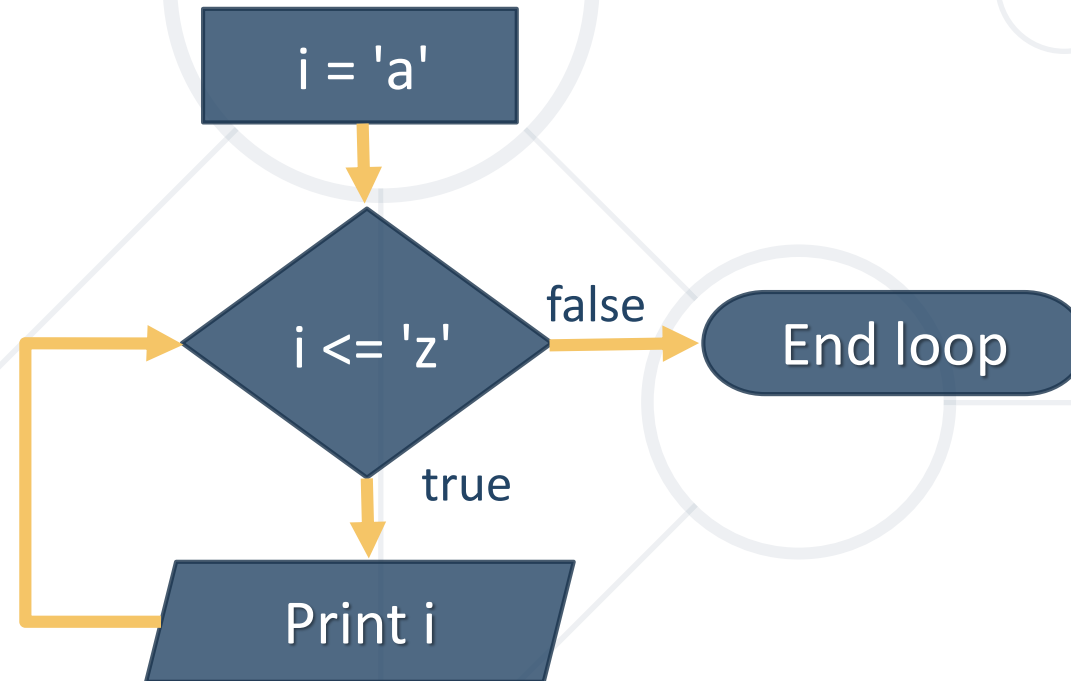
- В JavaScript можем да променяме типа на данните от число към символ и от символ към число, като използваме методите `fromCharCode()` и `charCodeAt()`
- Примери:
 - От символ към число и обратно:

```
let a = String.fromCharCode(67); // c  
let str = 'C';  
str.charCodeAt(0) // 67
```

Индекс на символа

Полученият резултат в число е ASCII стойността на символа

- Напишете програма, която:
 - Извежда буквите от латинската азбука: [a, z]



- Можем да вземем дължината на текст

```
let text = input.shift(); // въвеждаме SoftUni  
let size = text.length;  // 7
```

- Можем да вземем символ от текст по индекс

```
let text = input.shift(); // въвеждаме SoftUni  
let letter = text[4];     // U
```

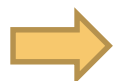
Сумиране на гласните букви - условие

- Напишете програма, която:
 - Получава като аргумент текст
 - Извежда сумата на гласните букви според таблицата по-долу:

а	е	и	о	у
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

```
let inputStr = input[0];  
let sum = 0;
```

```
for (let i = 0; i < inputStr.length; i++) {
```

```
  switch (inputStr[i]) {
```

```
    case 'a': sum += 1; break;
```

```
    case 'e': sum += 2; break;
```

```
    // TODO: Add cases for the other vowels.
```

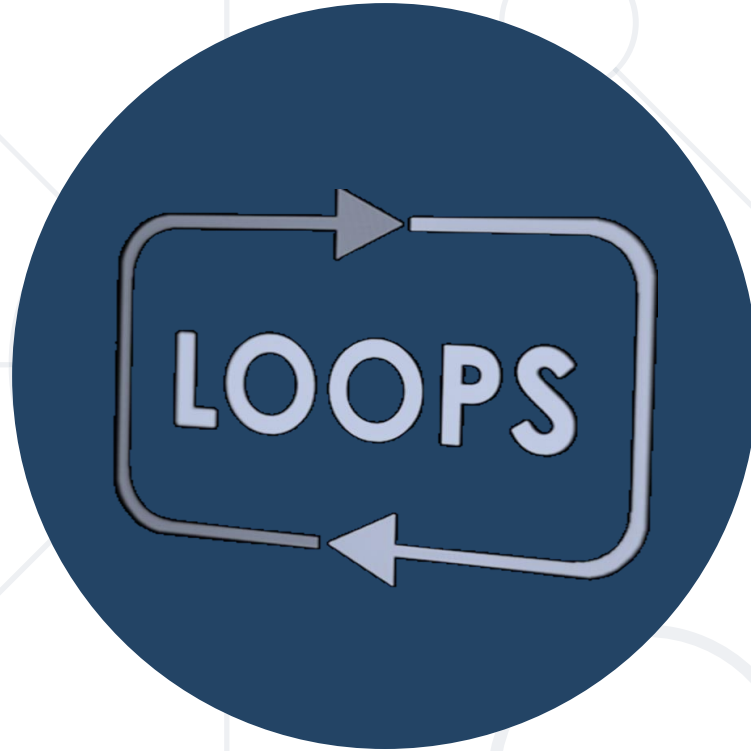
```
  }
```

```
}
```

```
console.log("Vowels sum = " + sum);
```

Можем да вземем
дължината на текста

Можем да вземем
символ по индекс i



Повторения на блокове код

Решаване на задачи в клас (лаб)



Техники за използване на for-цикли

Задачи с цикли

- Напишете програма, която:
 - Получава число n
 - Получава n последователни пъти числа и ги сумира
 - Извежда пресметнатата сума
- Примерен вход и изход:

2
10
20



30

3
-10
-20
-30

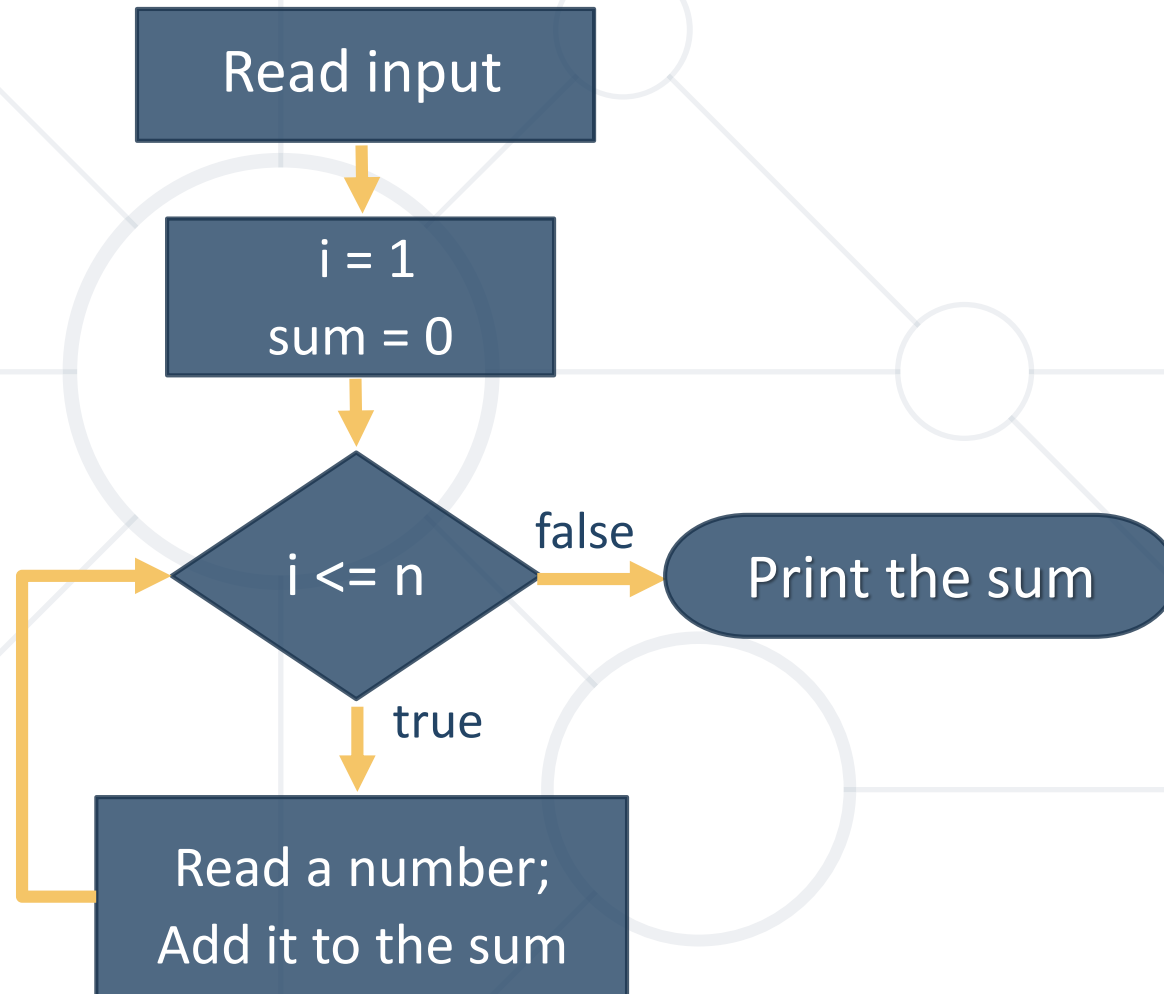
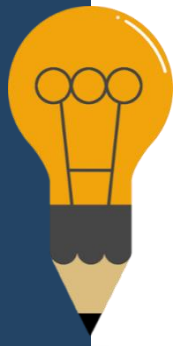


-60

4
45
-20
7
11



43



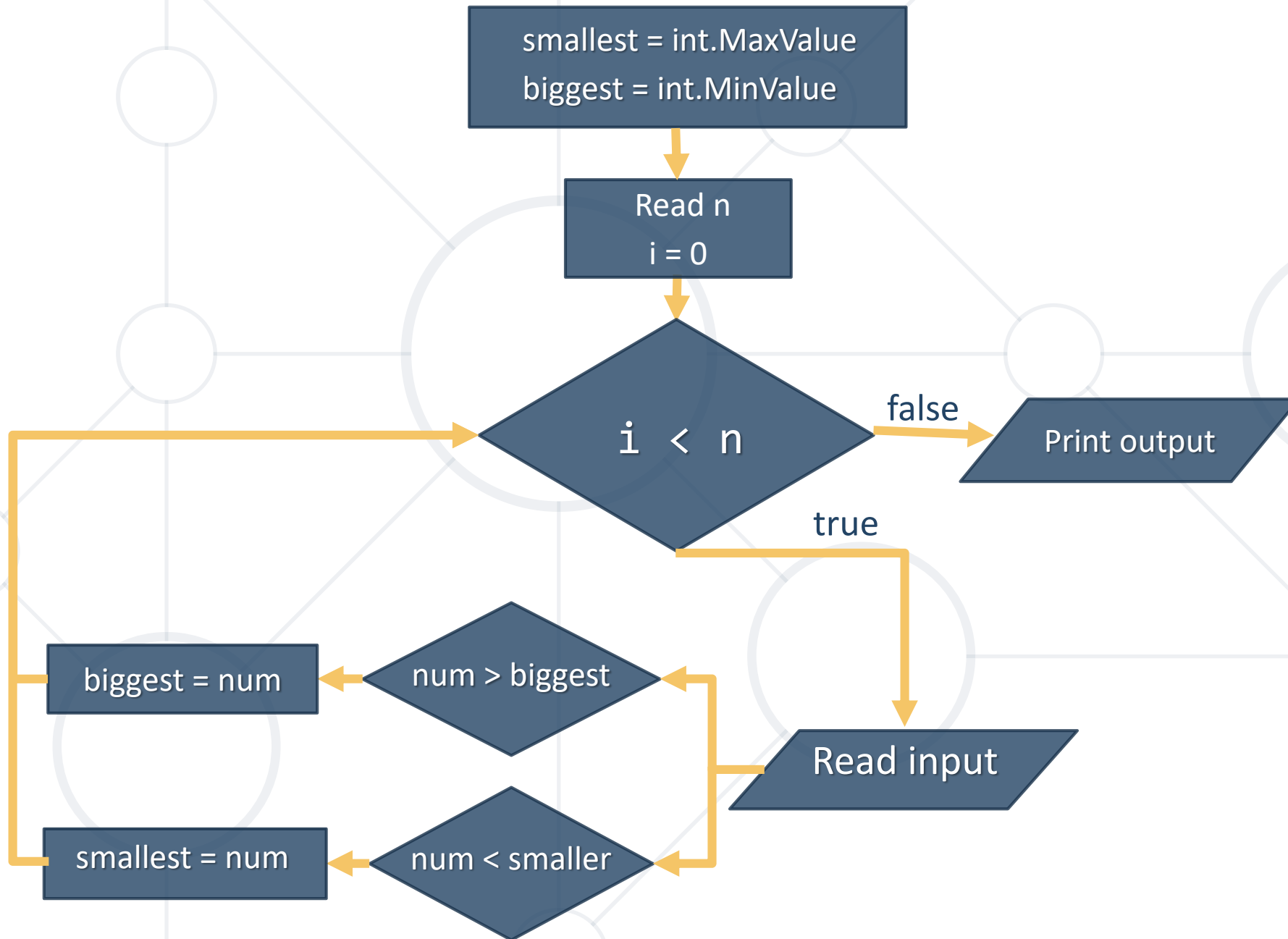
Редица цели числа - условие

- 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



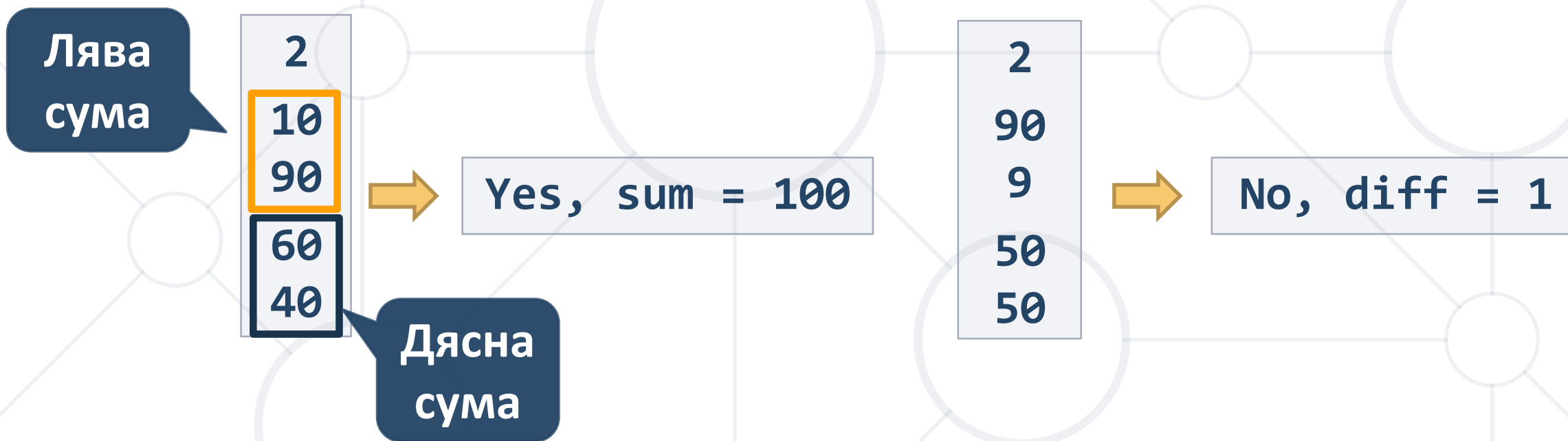
```
function numberSequence(input) {  
    let maxNumber = Number.MIN_SAFE_INTEGER;  
    let minNumber = Number.MAX_SAFE_INTEGER;  
    let n = Number(input.shift());  
    for (let i = 0; i < n; i++) {  
        let number = Number(input.shift());  
        if (number < minNumber) {  
            minNumber = number;  
        }  
        if (number > maxNumber) {  
            maxNumber = number;  
        }  
    }  
    //TODO: print the output  
}
```

1 3 7

- Напишете програма, която:
 - Получава цяло число n от потребителя
 - Взима 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

```
let n = Number(input[0]);
let leftSum = 0;
for (let i = 1; i <= n; i++){
    leftSum = leftSum + Number(input[i]);
}
// TODO: read and calculate the rightSum
if (leftSum == rightSum)
    console.log("Yes, sum = " + leftSum);
else
    let diff = Math.abs(rightSum - leftSum);
    console.log("No, diff = " + diff);
```

- Напишете програма, която:
 - Получава цяло число(n) от потребителя
 - Взима 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

```
let n = Number(input[0]);
let oddSum = 0;
let evenSum = 0;
for (let i = 1; i <= n; i++) {
  let element = Number(input[i]);
  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)

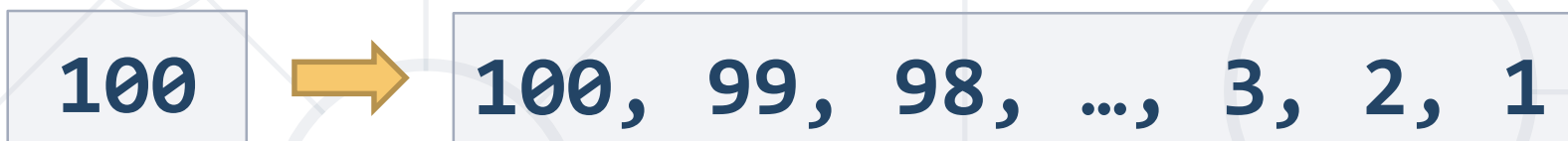


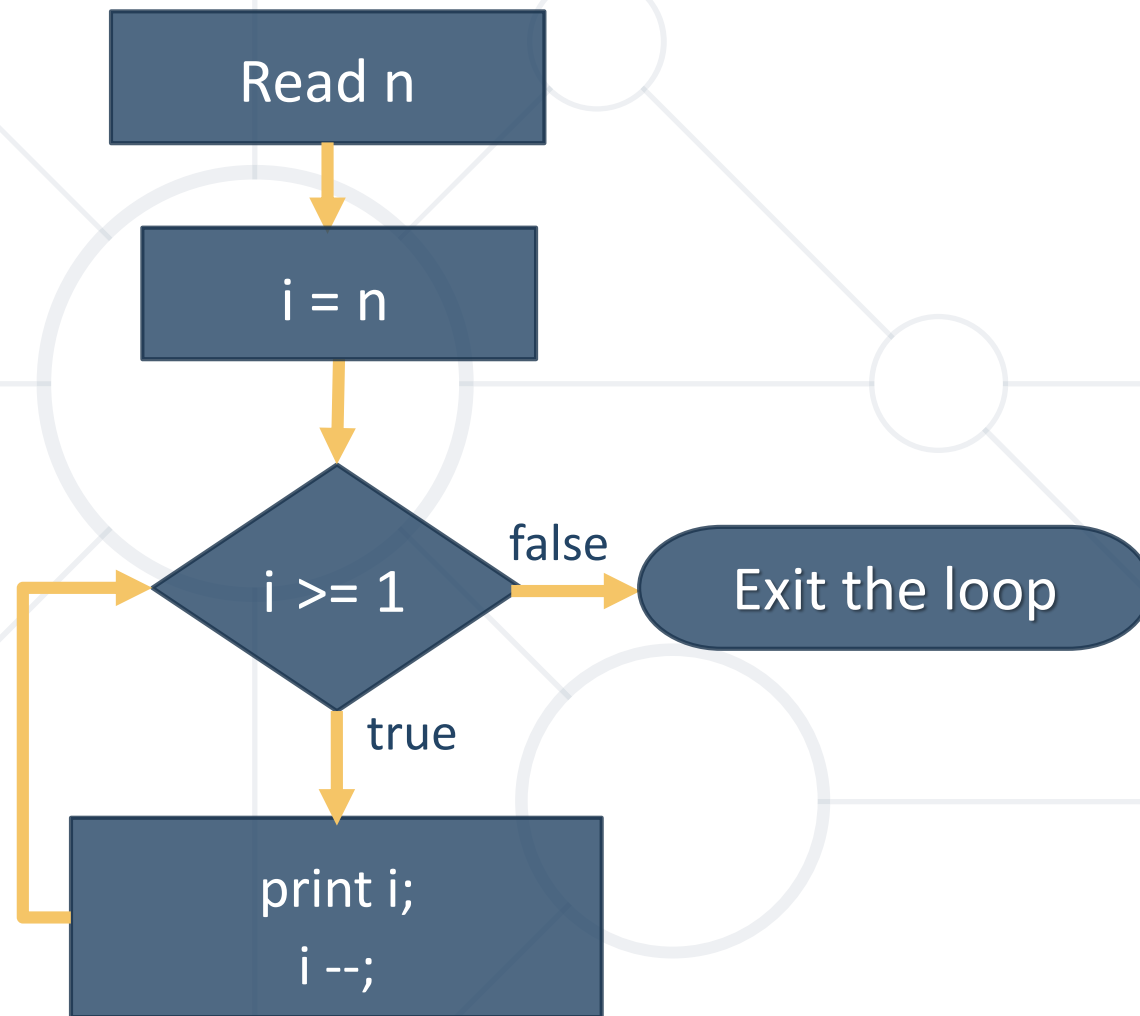
Работа с по-сложни for-цикли

Цикли със стъпка

Числата от N до 1 в обратен ред – условие

- Напишете програма, която:
 - Получава 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

```
let n = Number(input[0]);  
for (int i = n; i >= 1; i--) {  
    console.log(i);  
}
```

Намаляваща стъпка: -1

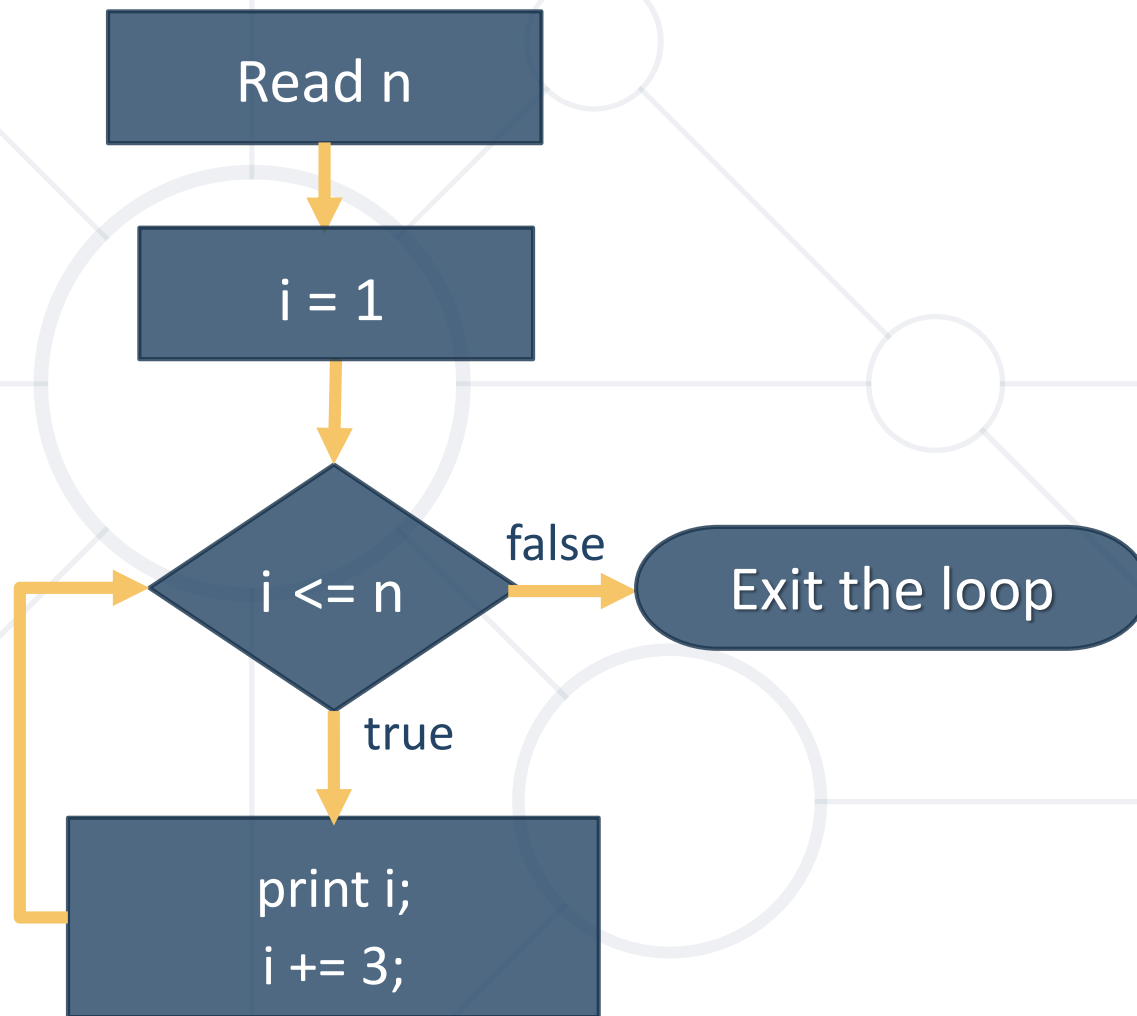
Обърнато условие: $i \geq 1$

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

Числата от 1 до N през 3 – условие

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





Числата от 1 до N през 3 – решение

```
let n = Number(input[0]);  
for (let i = 1; i <= n; i += 3) {  
    console.log(i);  
}
```

Задаване на
стъпка 3

Четни степени на 2 – условие

- Напишете програма, която:
 - Получава 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

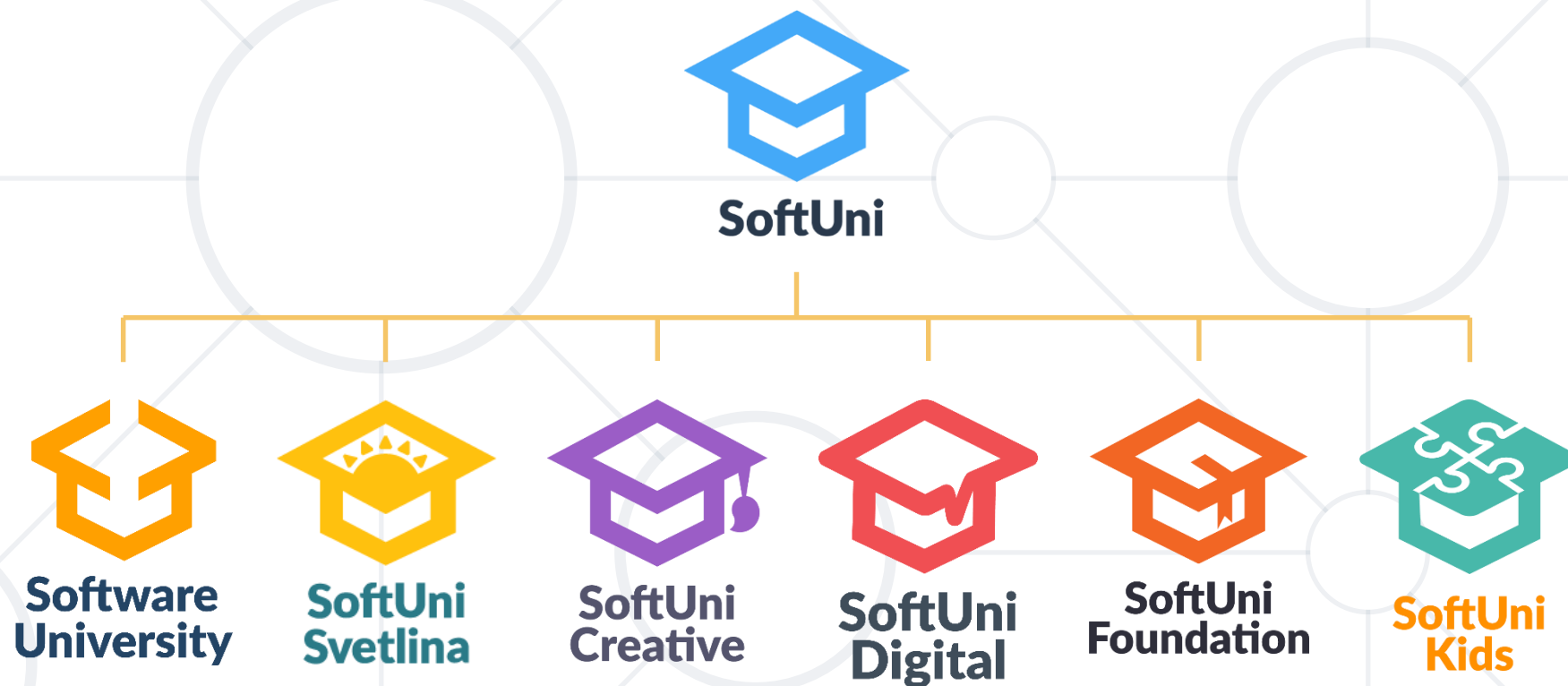
```
let n = Number(input[0]);  
let num = 1;  
for (let i = 0; i <= n; i += 2){  
    console.log(num);  
    num = num * 2 * 2;  
}
```

Ползваме
стъпка 2

- Символите, които използваме се представят като числа и са поместени в **ASCII** таблицата
- Можем да повтаряме блок код с **for**-цикъл
- Символите могат да се репрезентират като числа
- Можем да четем поредица от **n** числа
- Можем да вземем символ по индекс от текст
- Цикли със стъпка
 - Цикли с намаляваща стъпка



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

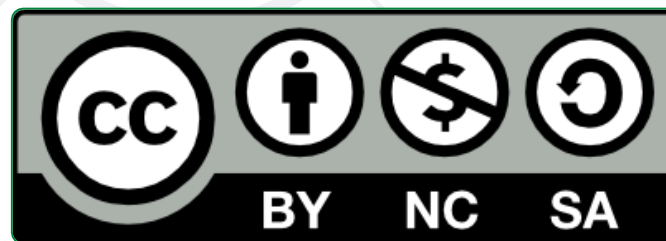


OneBit
SOFTWARE



WORLD
OF
MYTHS

- 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 JavaScript" от Светлин Наков и колектив с лиценз CC-BY-SA

Обучения в СофтУни

- 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



**Software
University**

