



Преговор

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

```
for (let i = 1; i<=3; ) {  
  console.log(i);  
}
```

123

Infinite loop

111

Compile time
error

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

```
for (let i = 1; i<=3; ) {  
  console.log(i);  
}
```

123

Infinite loop

111

Compile time
error

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

```
for(; ;) {  
    console.log("SoftUni");  
}
```

Infinite loop

Runtime error

Compile
time error

SoftUni

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

```
for(; ;) {  
    console.log("SoftUni");  
}
```

Infinite loop

Runtime error

Compile
time error

SoftUni

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

```
for (let i = 97; i < 100; i++) {  
  console.log(String.fromCharCode(i));  
}
```

a b c

Compile time
error

97 98 99

Infinite loop

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

```
for (let i = 97; i < 100; i++) {  
  console.log(String.fromCharCode(i));  
}
```

a b c

Compile time
error

97 98 99

Infinite loop

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

```
for(let i = 48; i < "A".charCodeAt(0); i++) {  
  console.log(String.fromCharCode(i));  
}
```

ASCII symbols:
0 ... 'A'

Infinite loop

0... 64

Compile time
error

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

```
for(let i = 48; i < "A".charCodeAt(0); i++) {  
  console.log(String.fromCharCode(i));  
}
```

ASCII symbols:
0 ... 'A'

Infinite loop

0... 64

Compile time
error

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

```
for (let i = 0; i < 2; i += 0.5) {  
  console.log(i);  
}
```

0, 0.5, 1, 1.5

0, 1

Infinite loop

Compile time
error

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

```
for (let i = 0; i < 2; i += 0.5) {  
  console.log(i);  
}
```

0, 0.5, 1, 1.5

0, 1

Infinite loop

Compile time
error

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

```
for (let i = 1; i <= 5; i++) {  
  if (i == 2 || i == 3) {  
    continue;  
  }  
  console.log(i);  
}
```

1 2 3 4 5

1 4 5

1 2 3 4

Compile time
error

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

```
for (let i = 1; i <= 5; i++) {  
  if (i == 2 || i == 3) {  
    continue;  
  }  
  console.log(i);  
}
```

1 2 3 4 5

1 4 5

1 2 3 4

Compile time
error

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

```
for (let i = 1; i <= 10; i++) {  
  console.log(i);  
  if (i % 10 == 5) {  
    break;  
  }  
}
```

1 2 3 4 5

1 2 3 4

Compile
time error

5 10

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

```
for (let i = 1; i <= 10; i++) {  
  console.log(i);  
  if (i % 10 == 5) {  
    break;  
  }  
}
```

1 2 3 4 5

1 2 3 4

Compile
time error

5 10

Работа с вложени цикли

Цикли със стъпка, по-сложни задачи



СофтУни

Преподавателски екип



Software
University



SoftUni
Foundation



Софтуерен университет

<http://softuni.bg>

sli.do

#pb-nov



Вложени цикли

По-сложни комбинаторни задачи

Пример – часовник (1)

Часовете се променят
когато минутите
надвишат 59

19:03

Докато минутите се
променят часовете
остават същите



Как може да си направим часовник с код?

Демо

Пример – часовник (2)

- Външният цикъл отговаря за часовете, а вътрешния за минутите

```
for (let h = 0; h <= 23; h++) {  
  for (let m = 0; m <= 59; m++) {  
    console.log(`${h}:${m}`);  
  }  
}
```

PROBLEMS	OUTPUT	DEBUG CONSOLE
	10:51	
	10:52	
	10:53	
	10:54	
	10:55	
	10:56	
	10:57	
	10:58	
	10:59	
	11:0	
	11:1	
	11:2	
	11:3	
	11:4	
	11:5	
	11:6	
	11:7	
	11:8	
	11:9	
	11:10	

```
for (let i = 0; i < n; i++)  
  for (let j = 0; j < n; j++)  
    ...
```

Имената на
итераторите трябва
да бъдат различни

- За всяка итерация на външния цикъл вложения се изпълнява **n** - на брой пъти



Таблица за умножение - условие

- Отпечатайте на конзолата таблицата за умножение за числата от 1 до 10
- Изход:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE
C:\Program Files\nodejs\node
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
3 * 1 = 3
```

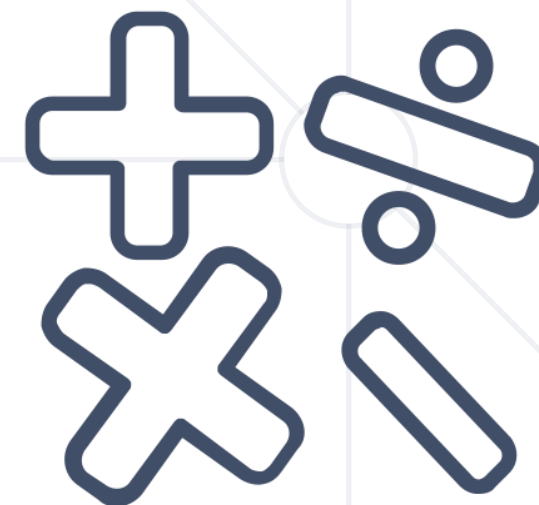


Таблица за умножение - решение

```
for (let x = 1; x <= 10; x++) {  
  for (let y = 1; y <= 10; y++) {  
    let product = x * y;  
    console.log(`${x} * ${y} = ${product}`);  
  }  
}
```


- За прекъсване на вложени цикли, използваме булеви променливи.

Външният цикъл ще се прекъсне, само ако стойността на `flag` бъде `true`

```
let flag = false;
for (let i = 0; i < n; i++)
  for (let j = 0; j < n; j++)
    if (condition)
      flag = true;
      break;
    if (flag)
      break;
```

- Напишете програма, която проверява всички възможни комбинации от двойка числа в даден интервал
 - Ако се намери комбинация, чийто **сбор от числата е равен** на дадено **магическо число** на изхода **се отпечатва съобщение** и програмата приключва изпълнение
 - Ако не се намери **ниито една комбинация**, отговаряща на условието се отпечатва **съобщение, че не е намерено**

Сума от две числа – условие (2)

- Примерен вход и изход:

1
10
5



Combination N:4 ($1 + 4 = 5$)

23
24
20



4 combinations - neither equals 20

Сума от две числа - решение

```
let startingNumber = Number(input.shift());
let finalNumber = Number(input.shift());
let magicNumber = Number(input.shift());
let combinations = 0;
let isFound = false;
for (let i = startingNumber; i <= finalNumber; i++)
  for (let j = startingNumber; j <= finalNumber; j++)
    combinations++;
    if (i + j === magicNumber)
      console.log(`Combination N:${combinations} (${i} + ${j} =
${magicNumber})`);
      isFound = true;
      break;
    if (isFound)
      break;
// Finish logic
```

Ако намерим
комбинация, прекъсваме
вътрешният цикъл

- Напишете програма, която изчислява **ASCII стойността** на няколко имена, като знаем че:
 - Името с **най-голяма стойност** е победител
 - Стойността на името е **сбора от ASCII стойностите** на всички **букви**
 - От конзолата ще се четат имена до получаването на команда **"STOP"**, след което трябва да се изпише:
"Winner is {името на победителя} – {стойността на името му}!".

Война на имена – условие (2)

- Примерен вход и изход:

Petar
Georgi
Stanimir
STOP



Winner is Stanimir – 839!

Ivo
Niki
Valio
Konstantin
STOP



Winner is Konstantin – 1065!

```
let input = inputArr.shift();
let maxCombination = 0;
let winner = "";
while (input !== "STOP")
    let currentSum = 0;
    for (let i = 0; i < input.Length; i++) {
        currentSum += input.charCodeAt(i);
    }
    if (currentSum > maxCombination)
        maxCombination = currentSum;
        winner = input;
    input = inputArr.shift();
console.log(`Winner is ${input} - ${maxCombination}!`);
```

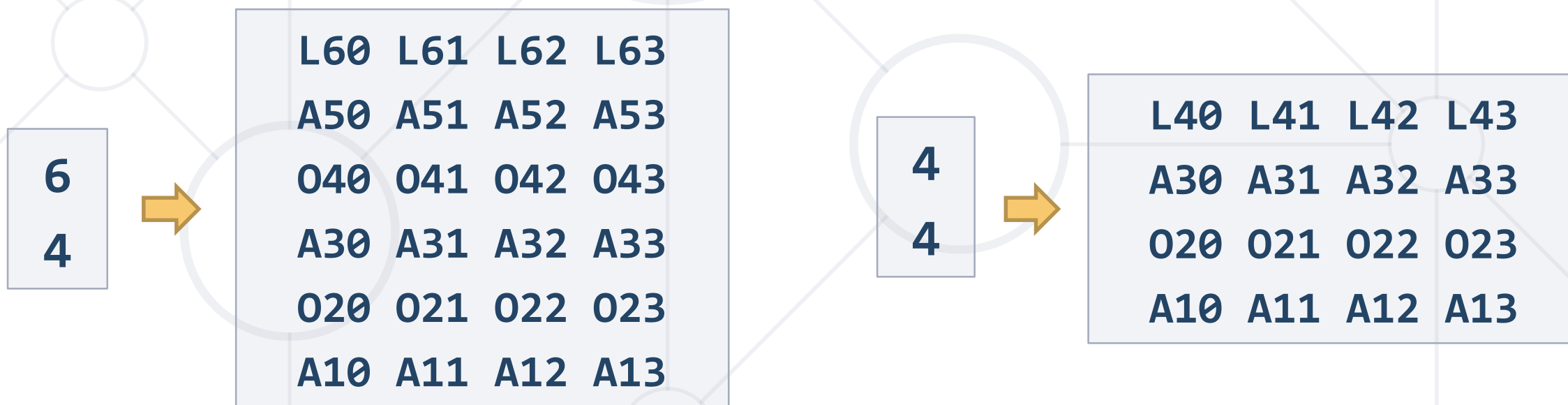
**Вложеният цикъл
обхожда името**

- Напишете програма, която извежда номерата на стаите в една сграда (в низходящ ред)
 - На всеки **четен** етаж има само **офиси**
 - На всеки **нечетен** етаж има само **апартаменти**
- Етажите се означават по следния начин:
 - Апартаменти: "А{номер на етажа}{номер на апартамента}"
 - Офиси: "О{номер на етажа}{номер на офиса}"
 - Номерата им винаги започват с 0



Сграда – условие (2)

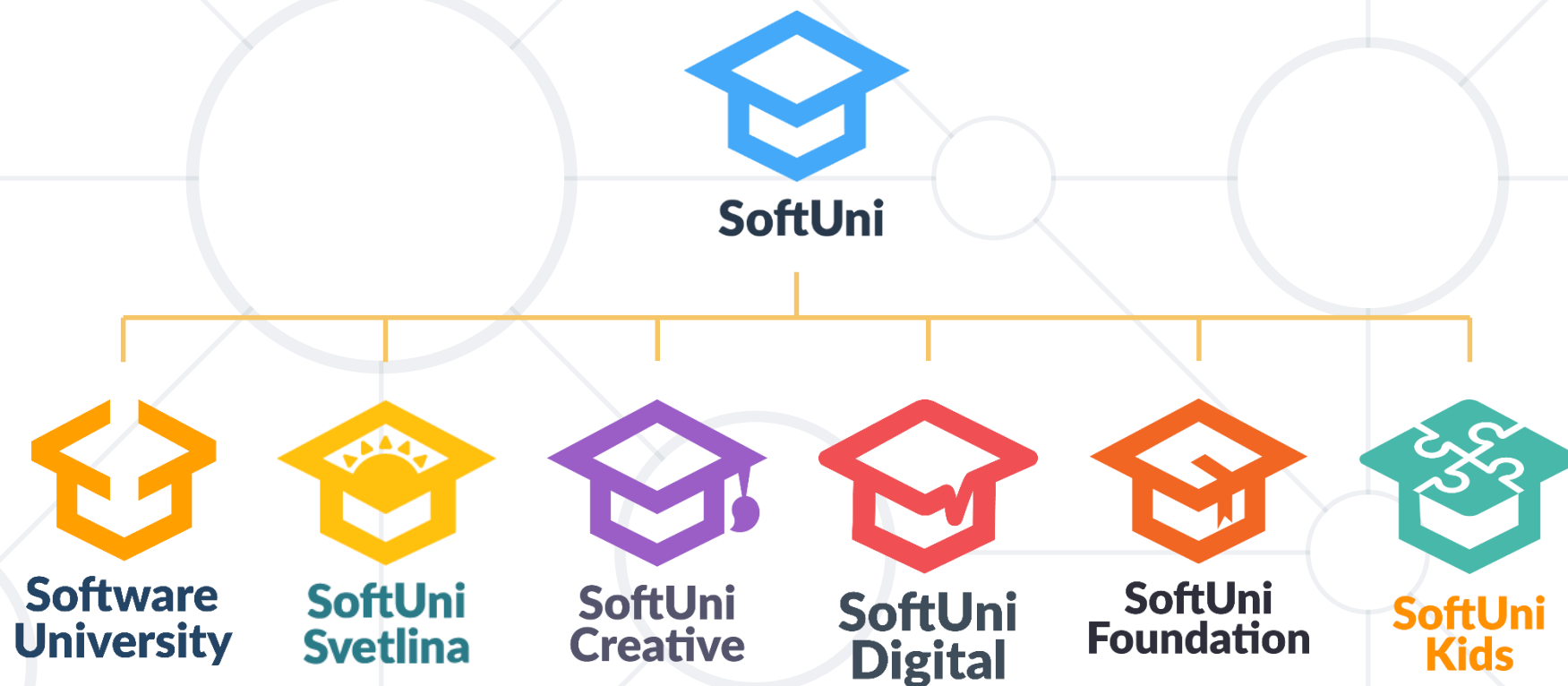
- На последният етаж винаги има големи апартаменти, които се означават с 'L', вместо с 'A'
- Ако има само един етаж, то има само големи апартаменти
- Примерен вход и изход:



```
let floors = Number(input[0]);
let rooms = Number(input[1]);
for (let i = floors; i >= 1; i--) {
  let printLine = "";
  for (let j = 0; j < rooms; j++) {
    if (i == floors)
      printLine += `L${i}${j} `;
    // TODO: print according to floor number
  }
  console.log(printLine);
}
```

Вложеният цикъл
итерира стаите

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.) с е 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 с JavaScript](#)" 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

