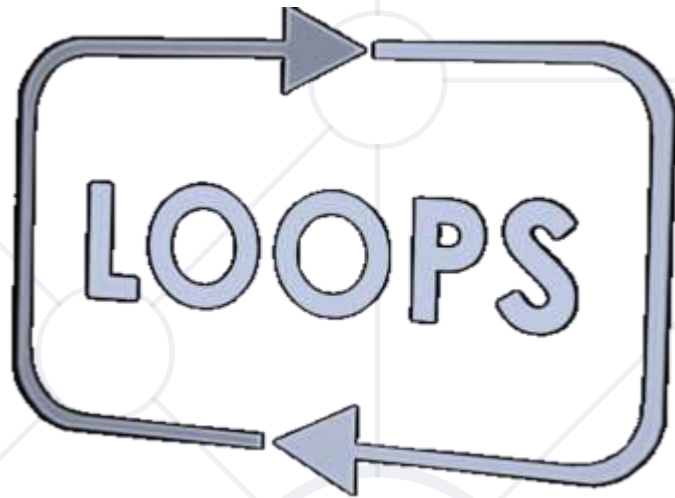


Повторения (цикли)

Прости повторения с For-цикъл



СофтУни

Преподавателски екип



SoftUni



Софтуерен университет

<http://softuni.bg>

Имате въпроси?

sli.do

#pb-jan

1. Преговор
2. Увеличаване и намаляване на стойността на променлива
3. Повторения на блокове код
4. Работа с по-сложни for-цикли
5. Работа с текст
6. Техники за използване на for-цикли





Преговор

1. Каква ще е стойността на променливата **a** след изпълнението на следната програма:

```
int a = 5;  
switch (a) {  
    case 5:  
        a = a + 1;  
        break;  
    default:  
        a = a + 2;  
        break;  
}
```

0

5

6

7

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

```
cout << (!(5 == 5) && (4 + 1 == 5)) << endl;
```

True

False

Runtime
error

Compile time
error

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

```
cout << (!(3 == 3) || (3 == 5)) << endl;
```

True

Runtime
error

False

Compile time
error

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

```
cout << (!(3 > 5) || (1 == 1)) << endl;
```

Compile time
error

Runtime
error

False

True

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

```
string role = "Administrators";  
string password = "SoftUni";  
if(role == "Administrator") {  
    if(password == "SoftUni") {  
        cout << "Welcome!" << endl;  
    }  
}
```

Welcome!

Runtime error

Compile time
error

No output



**Увеличаване и намаляване на стойността
на променливи**

- **Инкрементиране** - увеличаването на стойността на дадена променлива
 - Извършва се чрез оператори за инкрементиране: **префиксни** и **постфиксни**
 - Извършва се само върху променливи, които имат числена стойност

Пример	Име	Резултат
<code>++a</code>	Пре -инкрементация	Увеличава стойността с единица и връща <code>a</code>
<code>a++</code>	Пост -инкрементация	Връща <code>a</code> и увеличава стойността с единица

■ Пре-инкрементация

```
int a = 1;  
cout << ++a << endl;    // 2  
cout << a << endl;      // 2
```

Стойността на променливата а се увеличава с 1 и след това се принтира

■ Пост-инкрементация

```
int a = 1;  
cout << a++ << endl;    // 1  
cout << a << endl;      // 2
```

Първо се принтира променливата а и след това се увеличава с 1

- **Декрементиране** – намаляването на стойността на дадена променлива
 - Извършва се чрез оператори за декрементиране: **префиксни** и **постфиксни**
 - Извършва се само върху променливи, които имат числена стойност

Пример	Име	Резултат
--a	Пре -декрементация	Намалява стойността с единица и връща a
a--	Пост -декрементация	Връща a и намалява стойността с единица

- **Пре-**декрементация

```
int a = 1;  
cout << --a << endl; // 0  
cout << a << endl;  // 0
```

Стойността на променливата a се намалява с 1 и след това се принтира

- **Пост-**декрементация

```
int a = 1;  
cout << a-- << endl; // 1  
cout << a << endl;  // 0
```

Първо се принтира променливата a и след това се намалява с 1



Повторения на блокове код

Конструкция за for-цикъл

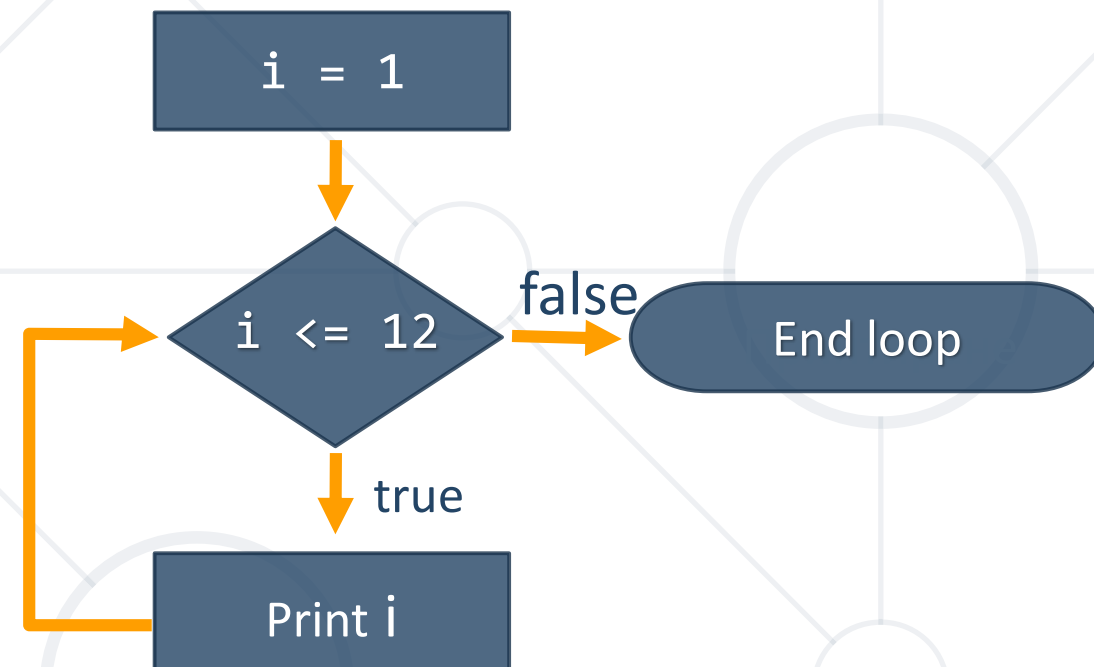
Какво е цикъл?

- Често ни се налага да **повтаряме** едно и също действие **многократно**
- Когато сме абитуренти броим до 12



Какво е цикъл? (2)

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



```
for (int i = 1; i <= 12; i += 1) {  
    cout << i << endl;  
}
```

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

Ключова дума за
конструкцията

Начална
стойност

Крайна
стойност

```
for (int i = 1; i <= 12; i += 1) {  
    cout << i << endl;  
}
```

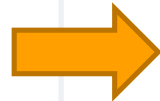
Стъпка

Тяло на цикъла: блок от код за
повторение

Числата от 1 до 100 - условие

- Напишете програма, която:
 - Отпечатва числата в диапазона **1** до **100**
 - Примерен вход и изход:

(няма вход)



1, 2, 3, ..., 100

Числата от 1 до 100 - решение

```
for (int i = 1; i <= 100; i += 1) {  
    cout << i << endl;  
}
```

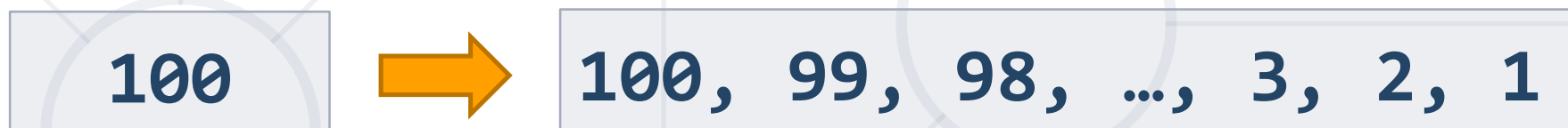


Работа с по-сложни For-цикли

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:



Read n

i = n

i >= 1

false

Exit the loop

true

print i;
i --;



Числата от N до 1 в обратен ред – решение

```
int n;  
cin >> n;  
for (int i = n; i >= 1; i--) {  
    cout << i << endl;  
}
```

Намаляваща стъпка: -1

Обърнато условие: $i \geq 1$

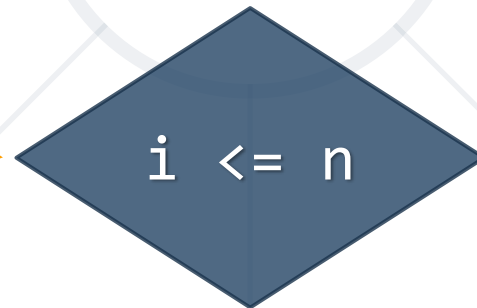
Числата от 1 до N през 3 – условие

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



Read n

`i = 1`



false

Exit the loop

true

`print i;`
`i +=3;`



Числата от 1 до N през 3 – решение

```
int n;  
cin >> n;  
for (int i = 1; i <= n; i += 3) {  
    cout << i << endl;  
}
```

Задаване на
стъпка 3



Работа с текст

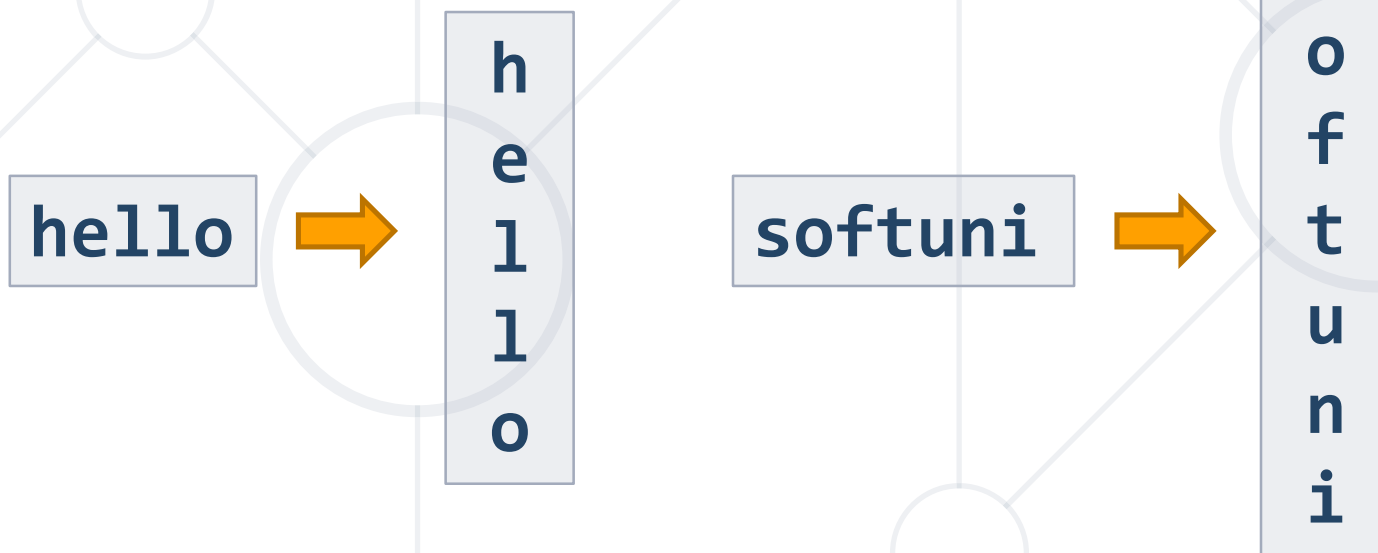
- Можем да вземем дължината на текст

```
string text;  
cin >> text;           // въвеждаме SoftUni  
int length = text.length(); // 7
```

- Можем да вземем символ от текст по индекс

```
string text;  
cin >> text;           // въвеждаме SoftUni  
char letter = text[4];  // U
```

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



Взимаме дължината
на текста

```
for (int i = 0; i < input.length(); i++) {  
    string letter;  
    cin >> input[i];  
    cout << letter << endl;  
}
```

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

```
string input; cin >> input;
int sum = 0;

for (int i = 0; i < input.length(); i++) {
    switch (input[i]) {
        case 'a': sum += 1; break;
        case 'e': sum += 2; break;
        // TODO: Add cases for the other vowels.
    }
}

cout << "Vowels sum = " << sum << endl;
```



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

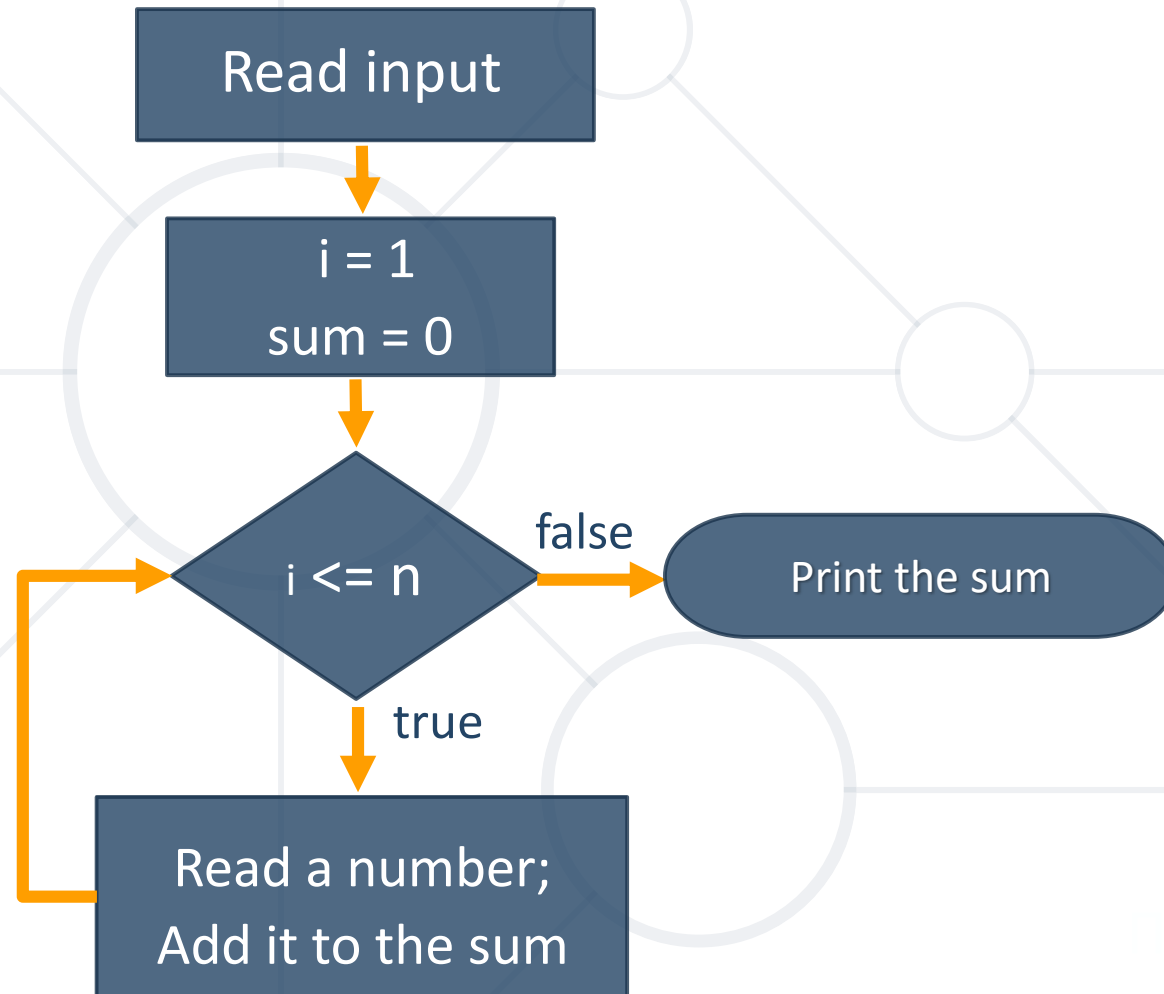


-60

4
45
-20
7
11



43



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

5
10
20
304
0
50

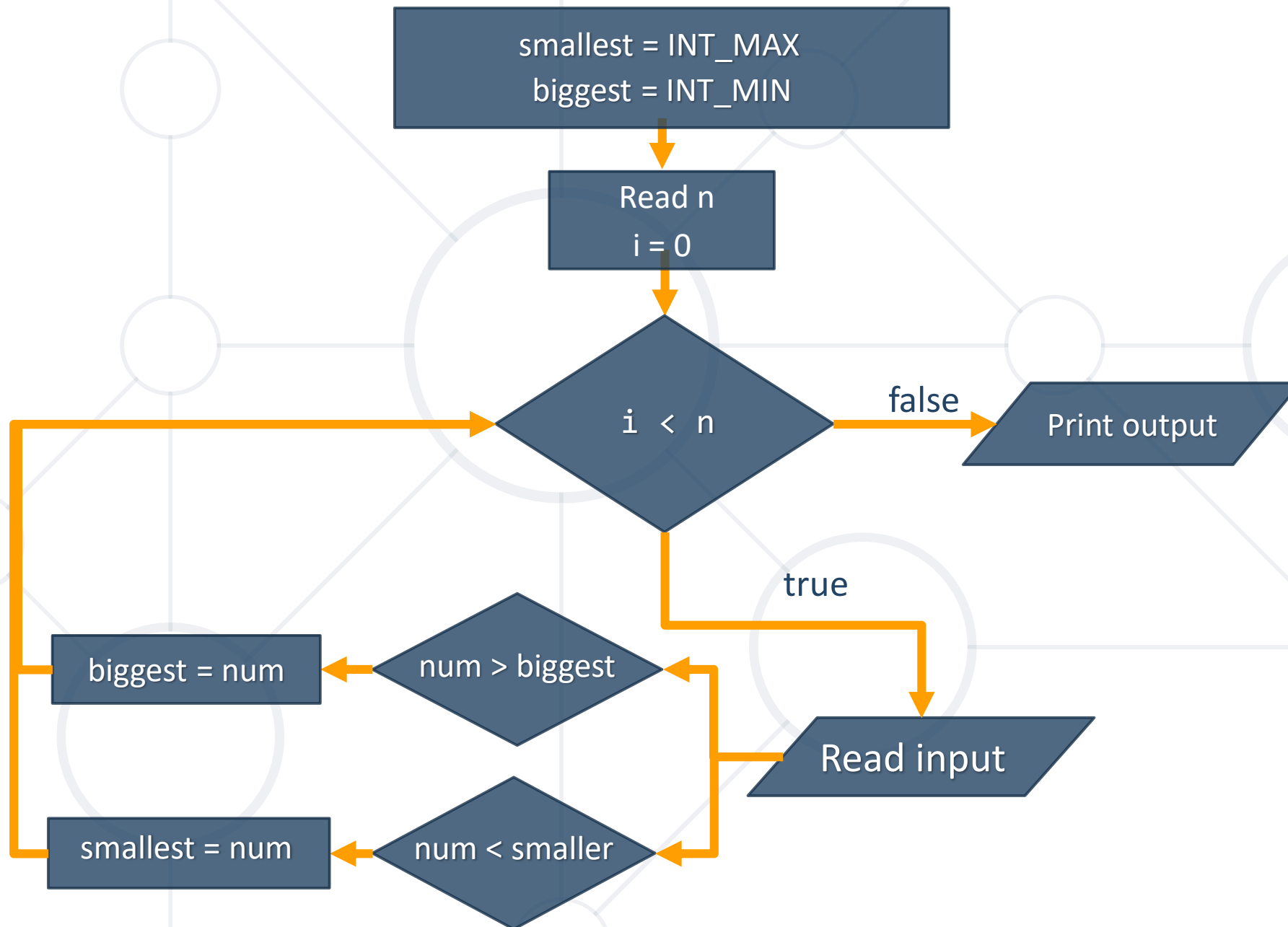


Max number: 304
Min number: 0

15
5
25
255
154
3



Max number: 255
Min number: 3



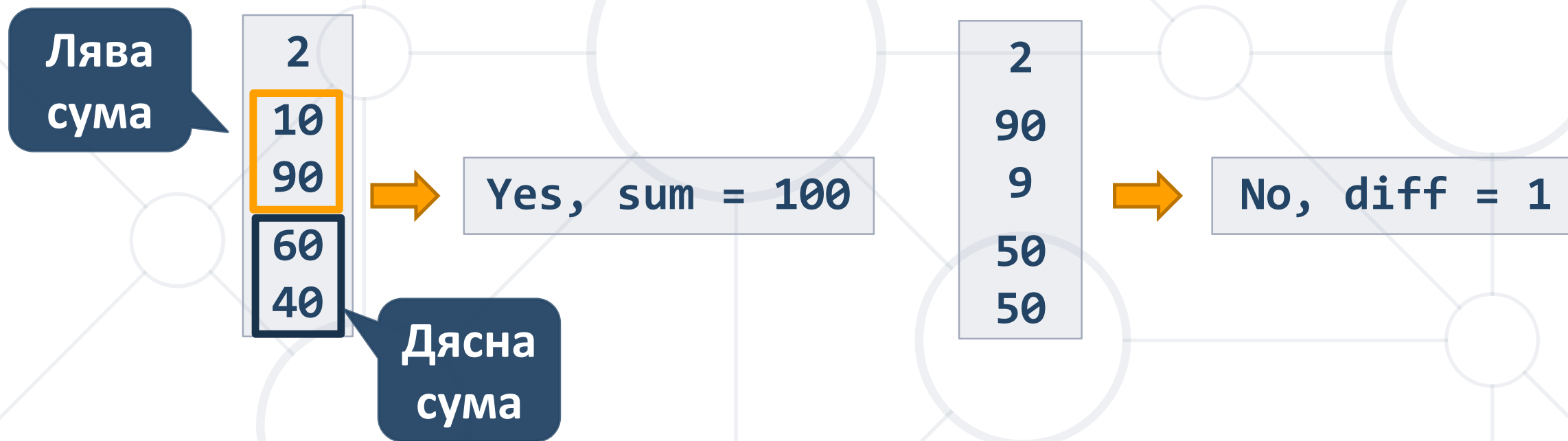
```
int smallest = INT_MAX;
int biggest = INT_MIN;
int n; cin >> n;
for (int i = 0; i < n; i++) {
    int num; cin >> num;
    if (num < smallest) smallest = num;
    if (num > biggest) biggest = num;
}
cout << "Max number: " << biggest << endl;
cout << "Min number: " << smallest << endl;
```

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 n; cin >> n;
int leftSum = 0;
for (int i = 1; i <= n; i++) {
    int currentNum; cin >> currentNum;
    leftSum = leftSum + currentNum;
}
// TODO: read and calculate the rightSum
if (leftSum == rightSum)
    cout << "Yes, sum = " << leftSum << endl;
else
    int diff = abs(rightSum - leftSum);
    cout << "No, diff = " << diff << endl;
```

- 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

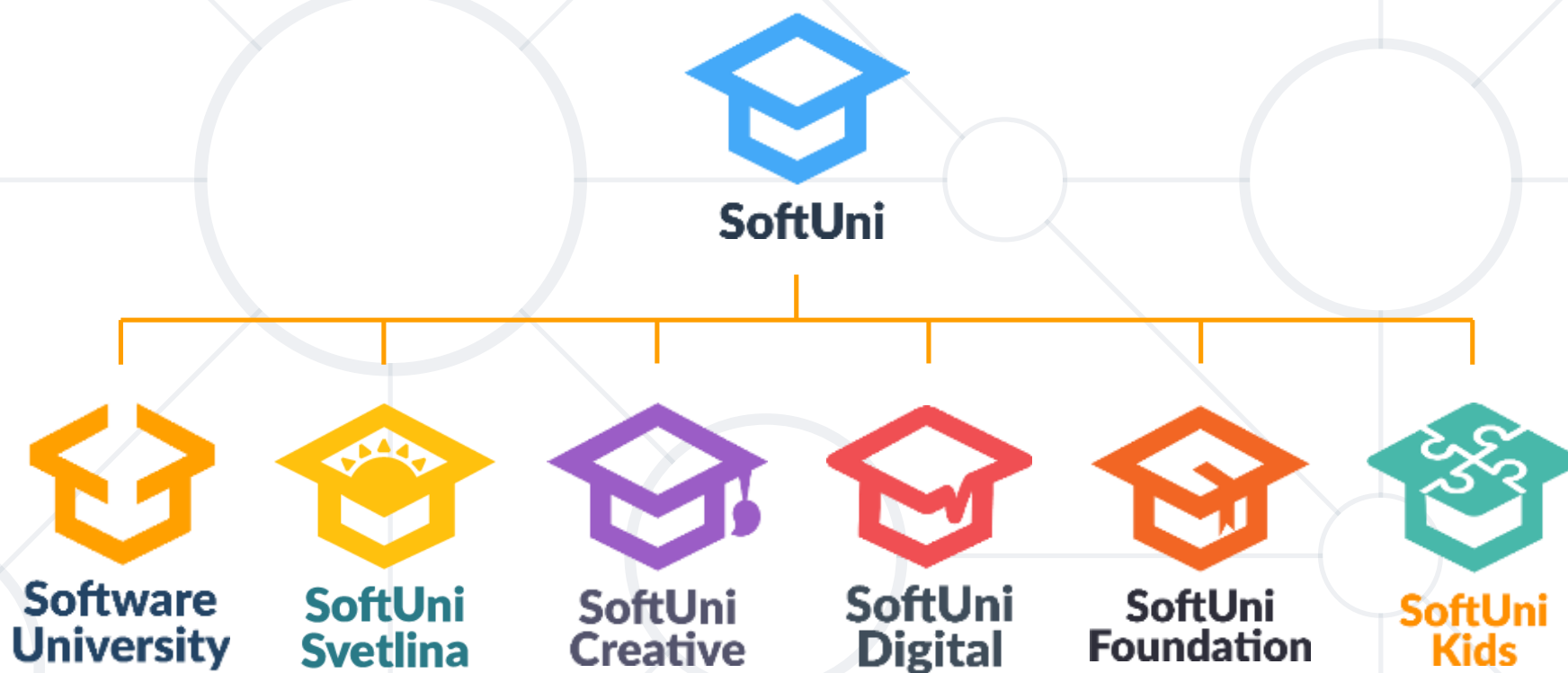
Решение: четна / нечетна сума

```
int n;  
cin >> n;  
int oddSum = 0;  
int evenSum = 0;  
for (int i = 1; i <= n; i++) {  
    int element; cin >> element;  
    if (i % 2 == 0) evenSum += element;  
    else oddSum += element;  
}  
  
// TODO: print the sum / difference
```

- Повторение на блок код с **for**-цикъл
- Цикли със стъпка
 - Цикли с увеличаваща стъпка
 - Цикли с намаляваща стъпка
- Достъпване на символ по индекс от текст



Въпроси?



- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни – <https://softuni.org>
- © Софтуерен университет – <https://softuni.bg>



- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
 - softuni.bg
- Фондация "Софтуерен университет"
 - softuni.foundation
- Софтуерен университет @ Facebook
 - facebook.com/SoftwareUniversity
- Дискуссионни форуми на СофтУни
 - forum.softuni.bg



Software University

