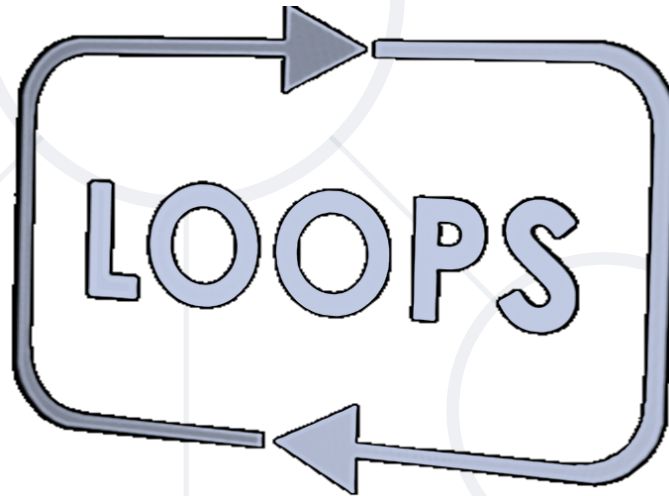


Повторения (цикли)

For-цикъл, While-цикъл, вложени цикли



СофтУни

Преподавателски екип



SoftUni

Софтуерен университет
<https://softuni.bg>

1. Повторения на блокове код
2. Работа с по-сложни **for-цикли**
3. Работа с текст
4. Конструкция и използване на **while-цикли**
5. Вложени цикли





Конструкция за For-цикъл

Какво е цикъл? (1)

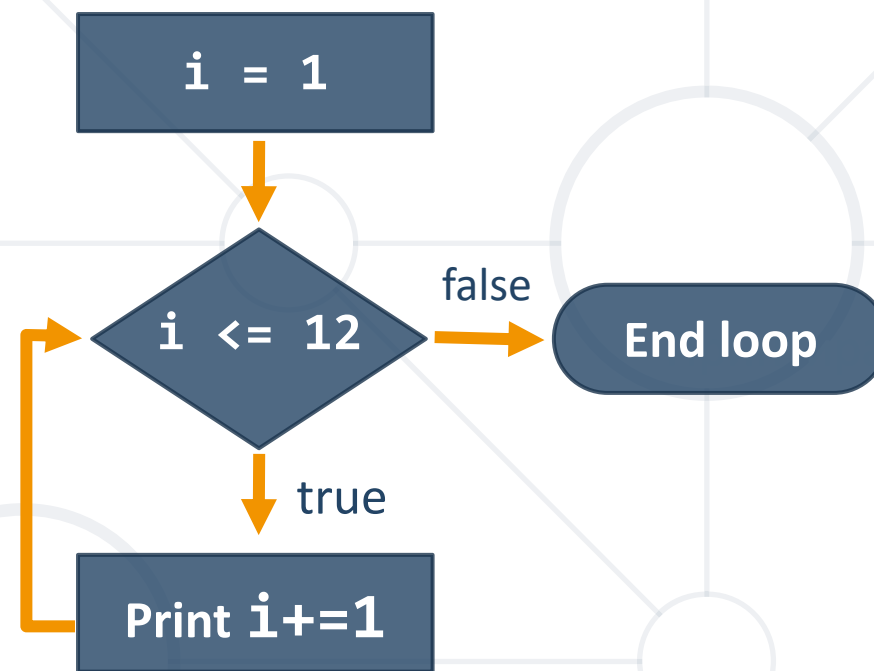
- Често ни се налага да **повтаряме** едно и също действие **многократно**
- Когато сме абитуренти броим до 12



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)
{
    Console.WriteLine(i);
}
```



- 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)
{
    Console.WriteLine(i);
}
```

Стъпка

Тяло на цикъла: блок от код за
повторение



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:

100 → 100, 99, 98, ..., 3, 2, 1

Read n

$i = n$

$i \geq 1$

false

Exit the loop

true

print i;
 $i--;$



Числата от N до 1 в обратен ред – решение

Обърнато условие: $i \geq 1$

```
int n = int.Parse(Console.ReadLine());  
for (int i = n;  $i \geq 1$ ;  $i--$ )  
{  
    Console.WriteLine(i);  
}
```

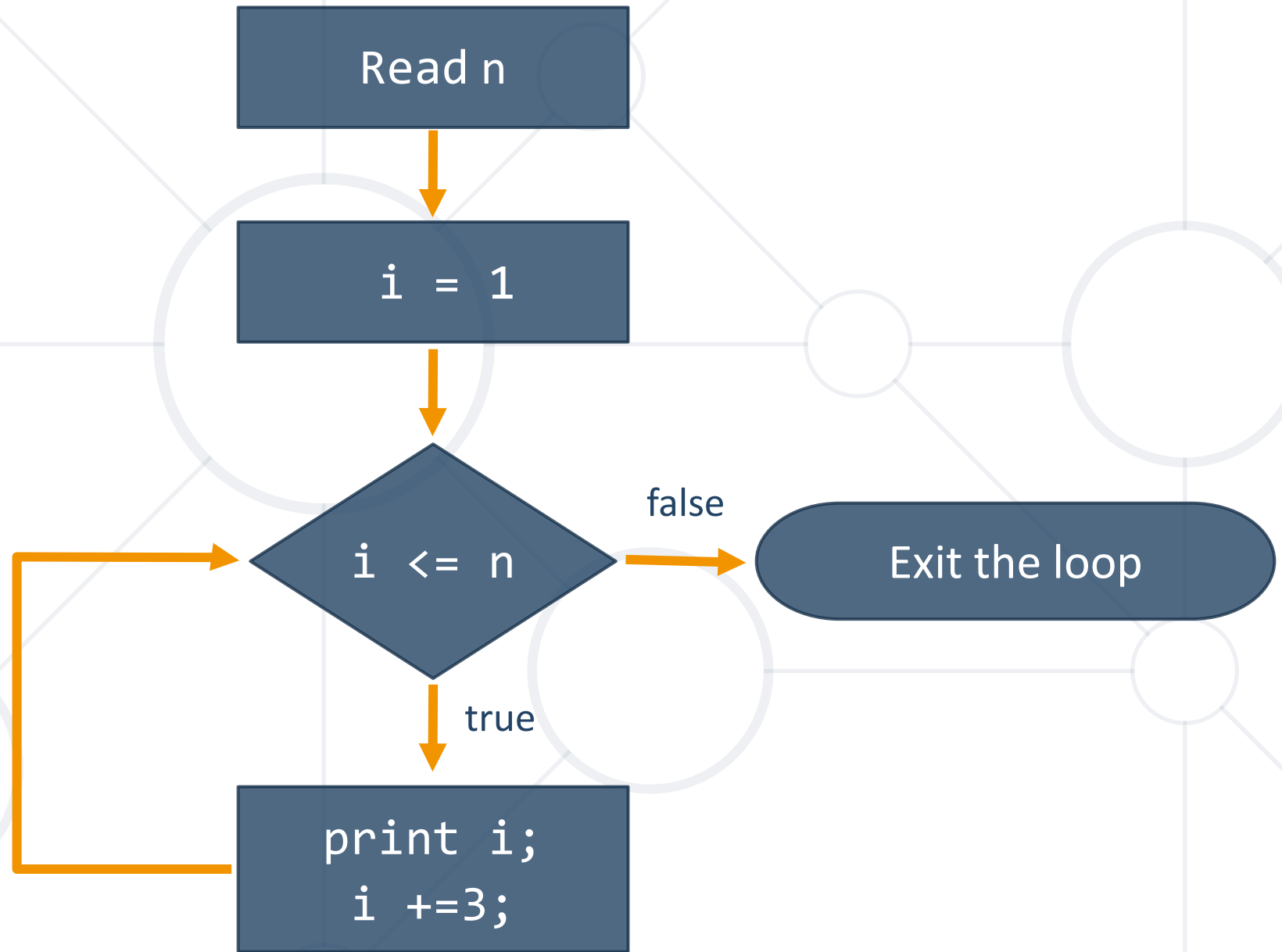
Намаляваща стъпка: -1

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/Practice/Index/3157#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:





Числата от 1 до N през 3 – решение

```
int n = int.Parse(Console.ReadLine());  
for (int i = 1; i <= n; i += 3)  
{  
    Console.WriteLine(i);  
}
```

Задаване на
стъпка 3

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/Practice/Index/3157#2>

Четни степени на 2 – условие

- Напишете програма, която:
 - Прочита цяло число **n**
 - Отпечатва четните степени на **2** до **2ⁿ**: $2^0, 2^2, 2^4, 2^6, 2^8, \dots, 2^n$
- Примерен вход и изход:

10 → 1, 4, 16, ..., 1024

7 → 1, 4, 16, 64

Четни степени на 2 – решение

```
int n = int.Parse(Console.ReadLine());  
int num = 1;  
for (int i = 0; i <= n; i += 2)  
{  
    Console.WriteLine(num);  
    num = num * 2 * 2;  
}
```

Ползваме
стъпка 2

- 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

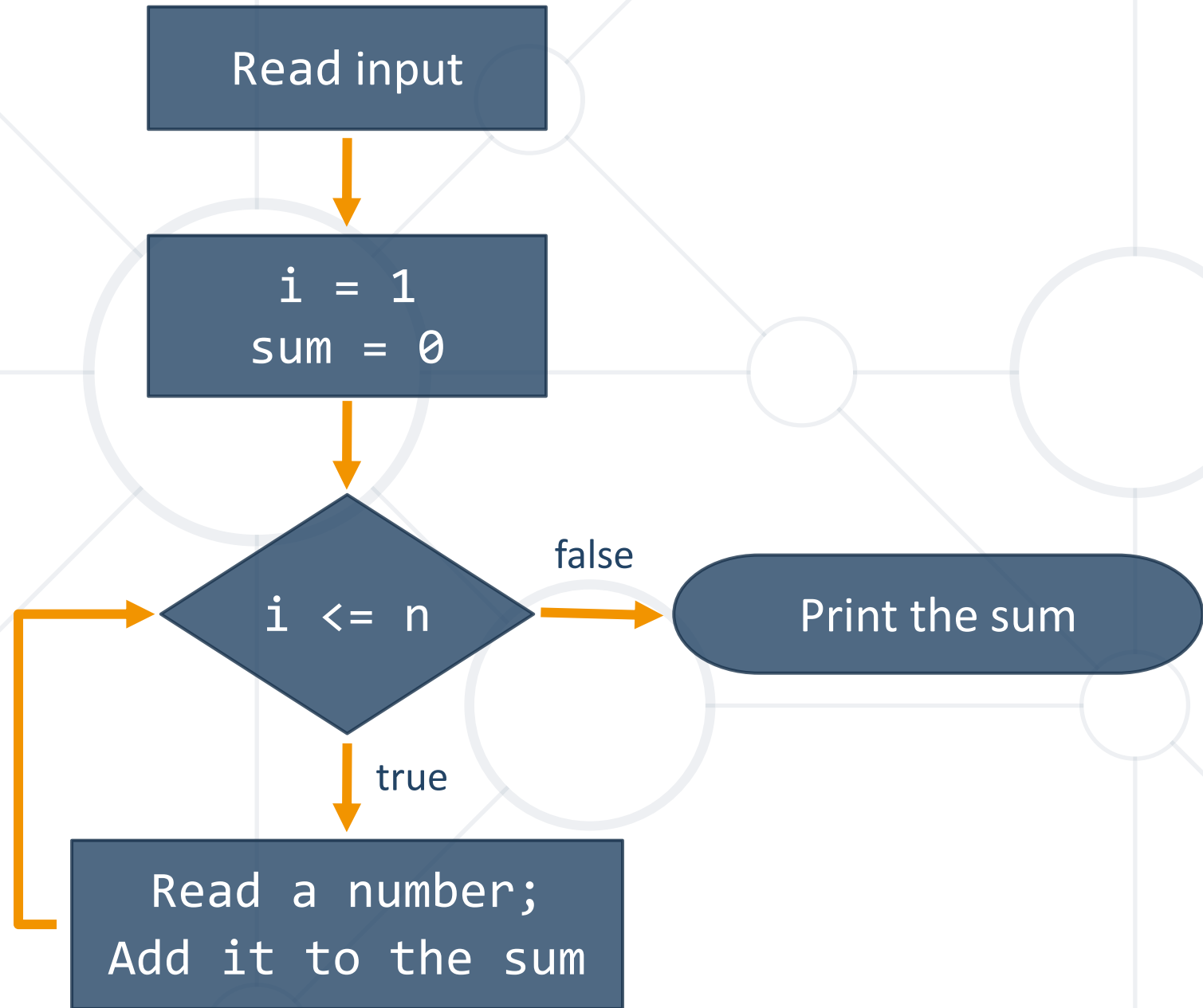


-60

4
45
-20
7
11



43



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

5
10
20
304
0
50

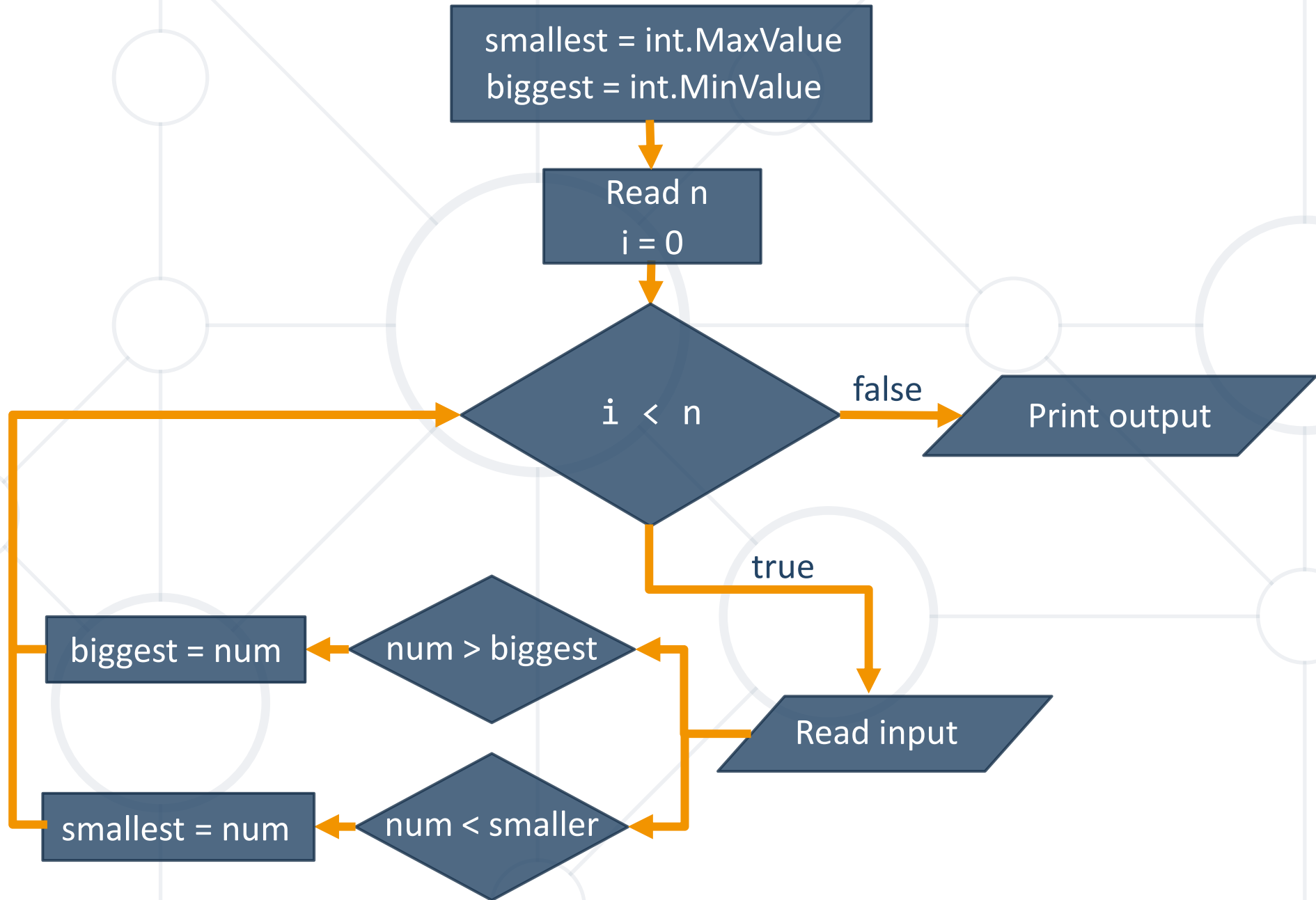


Max number: 304
Min number: 0

5
15
20
350
20
1



Max number: 350
Min number: 1



```
int smallest = int.MaxValue;
int biggest = int.MinValue;
int n = int.Parse(Console.ReadLine());

for (int i = 0; i < n; i++)
{
    int num = int.Parse(Console.ReadLine());
    if (num < smallest) smallest = num;
    if (num > biggest) biggest = num;
}

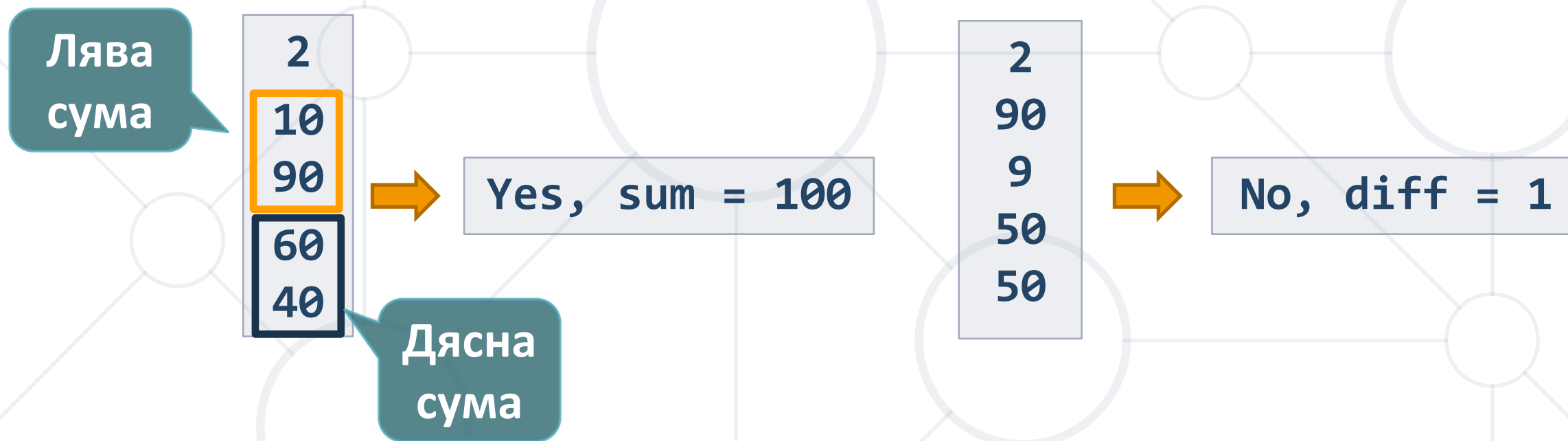
Console.WriteLine($"Max number: {biggest}");
Console.WriteLine($"Min number: {smallest}");
```

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 (2)

- Примерен вход и изход:



Лява и дясна сума – решение

```
int n = int.Parse(Console.ReadLine());
int leftSum = 0;
for (int i = 1; i <= n; i++)
{
    leftSum = leftSum + int.Parse(Console.ReadLine());
}
// TODO: read and calculate the rightSum
if (leftSum == rightSum)
    Console.WriteLine("Yes, sum = " + leftSum);
else
    int diff = Math.Abs(rightSum - leftSum);
    Console.WriteLine("No, diff = " + diff);
```

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/Practice/Index/3157#8>

- 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 (2)

- 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

Четна / нечетна сума – решение

```
int n = int.Parse(Console.ReadLine());
int oddSum = 0;
int evenSum = 0;
for (int i = 1; i <= n; i++)
{
    int element = int.Parse(Console.ReadLine());
    if (i % 2 == 0) evenSum += element;
    else oddSum += element;
}

// TODO: print the sum / difference
```



Работа с текст

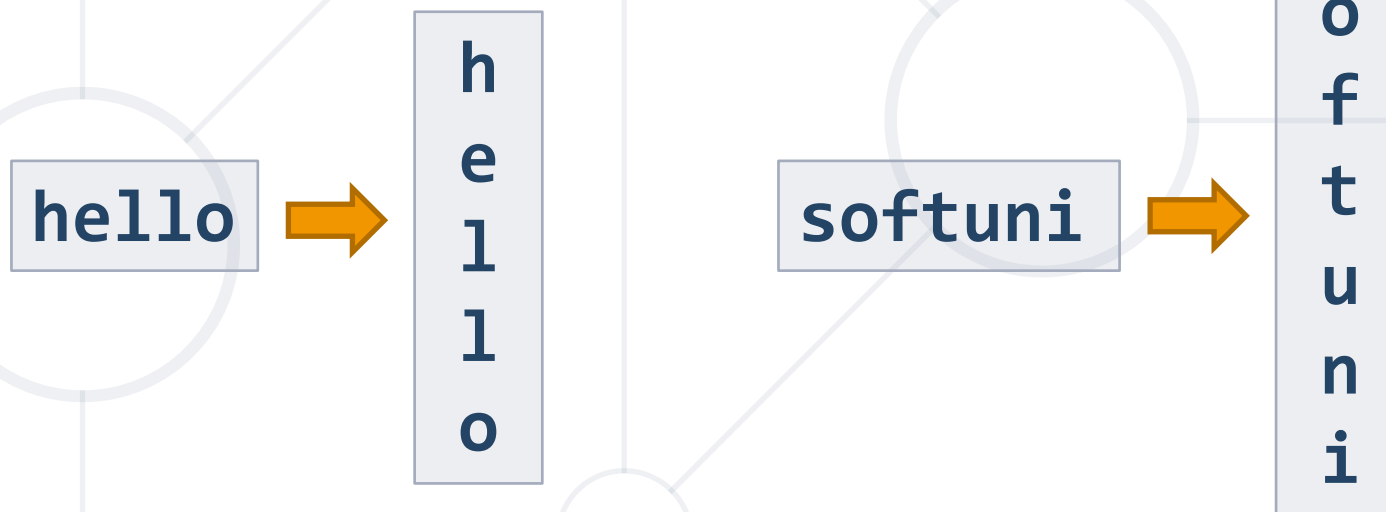
- Можем да вземем **дължината на текст**

```
string text = "SoftUni"  
int length = text.Length;    // 7
```

- Можем да вземем **СИМВОЛ** от текст по **ИНДЕКС**

```
string text = "SoftUni"  
char letter = text[4];    // u
```

- Напишете програма, която
 - Чете текст(**стринг**)
 - Печата всеки **СИМВОЛ** от текста на отделен ред
- Примерен вход и изход:



```
string input = Console.ReadLine();  
for (int i = 0; i < input.Length; i++)  
{  
    Console.WriteLine(input[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	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)

Сумиране на гласни букви – решение

```
string input = Console.ReadLine();
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.
    }
}
Console.WriteLine("Vowels sum = " + sum);
```

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/Practice/Index/3157#5>

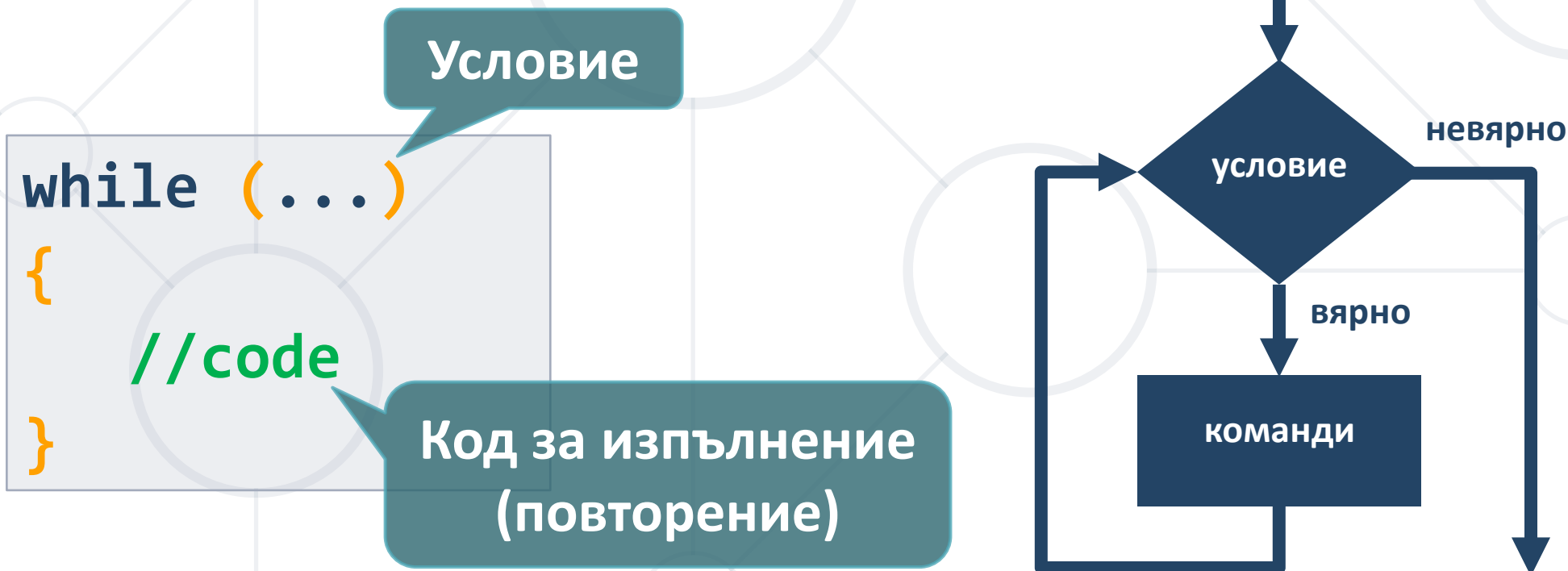
A background network diagram consisting of several light gray circles of varying sizes connected by thin gray lines. The circles are arranged in a non-uniform pattern, with some having more connections than others, creating a web-like structure.

while

**Повторение докато е вярно дадено
условие**

Повторения (цикли) – While-цикъл

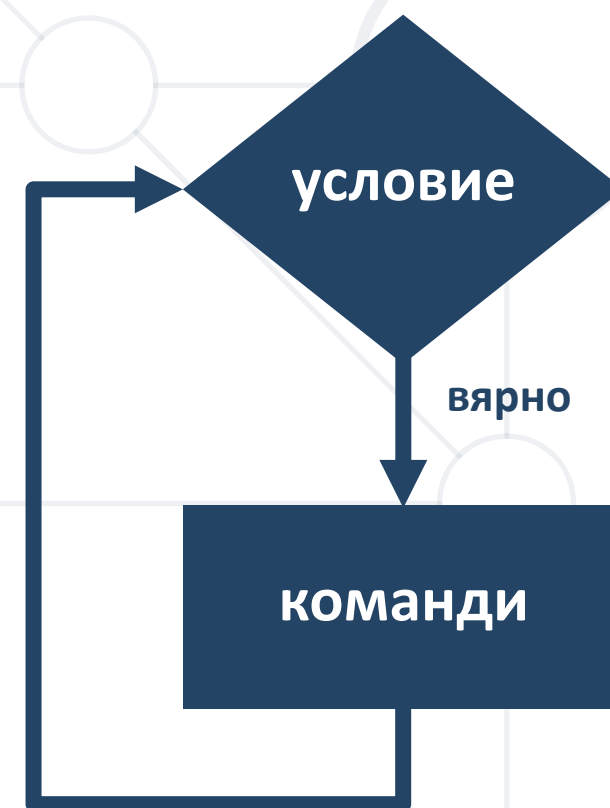
- В програмирането често се налага да изпълним блок с команди няколко пъти
 - За целта използваме цикли – **while**, **for** и други



- **Безкраен цикъл** – повтаряне на блок от код безкраен брой пъти:

Условието е винаги
вярно

```
while (true)
{
    Console.WriteLine("Infinite loop");
}
```



- Оператор **break** – прекъсва цикъла
- Не може да съществува самостоятелно извън цикъл

```
while (true)
{
    Console.WriteLine("Infinite loop");
    if (...)
    {
        break;
    }
}
```

Условие за прекъсване на
цикъла

While-цикъл – пример

Условие за прекратяване
на повторението

```
int a = 5;  
while (true)  
{  
    if (a > 10) {  
        break;  
    }  
    Console.WriteLine("a = " + a);  
    a++;  
}
```



```
a = 5  
a = 6  
a = 7  
a = 8  
a = 9  
a = 10  
Press any key to continue . . .
```

- Напишете програма, която:
 - **Чете** от потребителя текст(низ)
 - Приключва четенето когато получи командата "**Stop**"
- Примерен вход и изход:



```
Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText  
Stop  
AfterStop
```

```
Nakov  
SoftUni  
Sofia  
Bulgaria  
SomeText
```

```
while (true)
{
    string input = Console.ReadLine();
    if (input == "Stop")
    {
        break;
    }
    Console.WriteLine(input);
}
```

- Напишете програма, която:
 - Първоначално прочита **потребителско име** и **парола** на потребителски профил
 - Прочита **парола** за вход и проверява дали е **коректна**
 - При:
 - Невалидна парола, прочита нова
 - При коректно въведена парола, прекратява изпълнение


```
string username = Console.ReadLine();  
string password = Console.ReadLine();  
  
string input = Console.ReadLine();  
while (input != password)  
{  
    input = Console.ReadLine();  
}  
  
Console.WriteLine($"Welcome: {username}!");
```

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/Practice/Index/3157#12>

- Напишете програма, която:
 - Чете от потребителя **цели числа**
 - Приключва четенето когато получи **сума равна на първоначално въведеното число**
 - Извежда **сумата** на всички **прочетени числа**
- Примерен вход и изход:

100
10
20
30
45



105

20
1
2
3
4
5
6



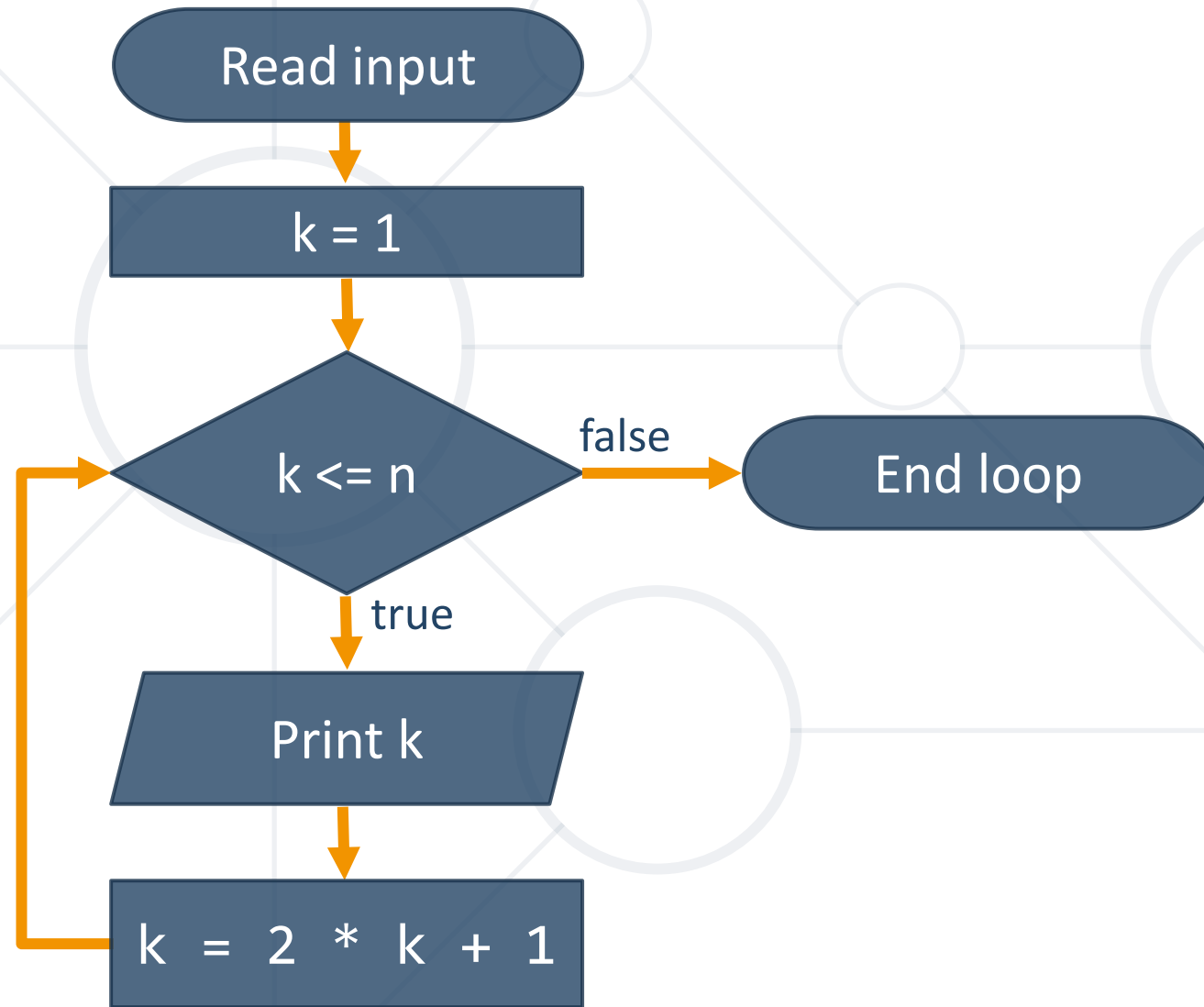
21

```
int n = int.Parse(Console.ReadLine());  
int sum = 0;  
  
while (sum < n)  
{  
    int currentNum =int.Parse(Console.ReadLine());  
    sum += currentNum;  
}  
Console.WriteLine(sum);
```

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** $\times 2 + 1$

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



Редица числа $2K+1$ – решение

```
int n = int.Parse(Console.ReadLine());  
int k = 1;  
while (k <= n)  
{  
    Console.WriteLine(k);  
    k = 2 * k + 1;  
}
```

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

- Напишете програма, която:
 - Чете **n** – на брой числа, които представляват вноски по банкова сметка до получаване на командата "**NoMoreMoney**"
 - При всяка вноска принтира: "**Increase: {сумата}**"
 - Ако се въведе отрицателно число да се изпише "**Invalid operation!**" и програмата да приключи
 - Накрая на програмата трябва да се изпише: "**Total: {общата сума в сметката}**"

Баланс на сметка – условие(2)

- Примерен вход и изход:

5.51
69.42
100
NoMoreMoney

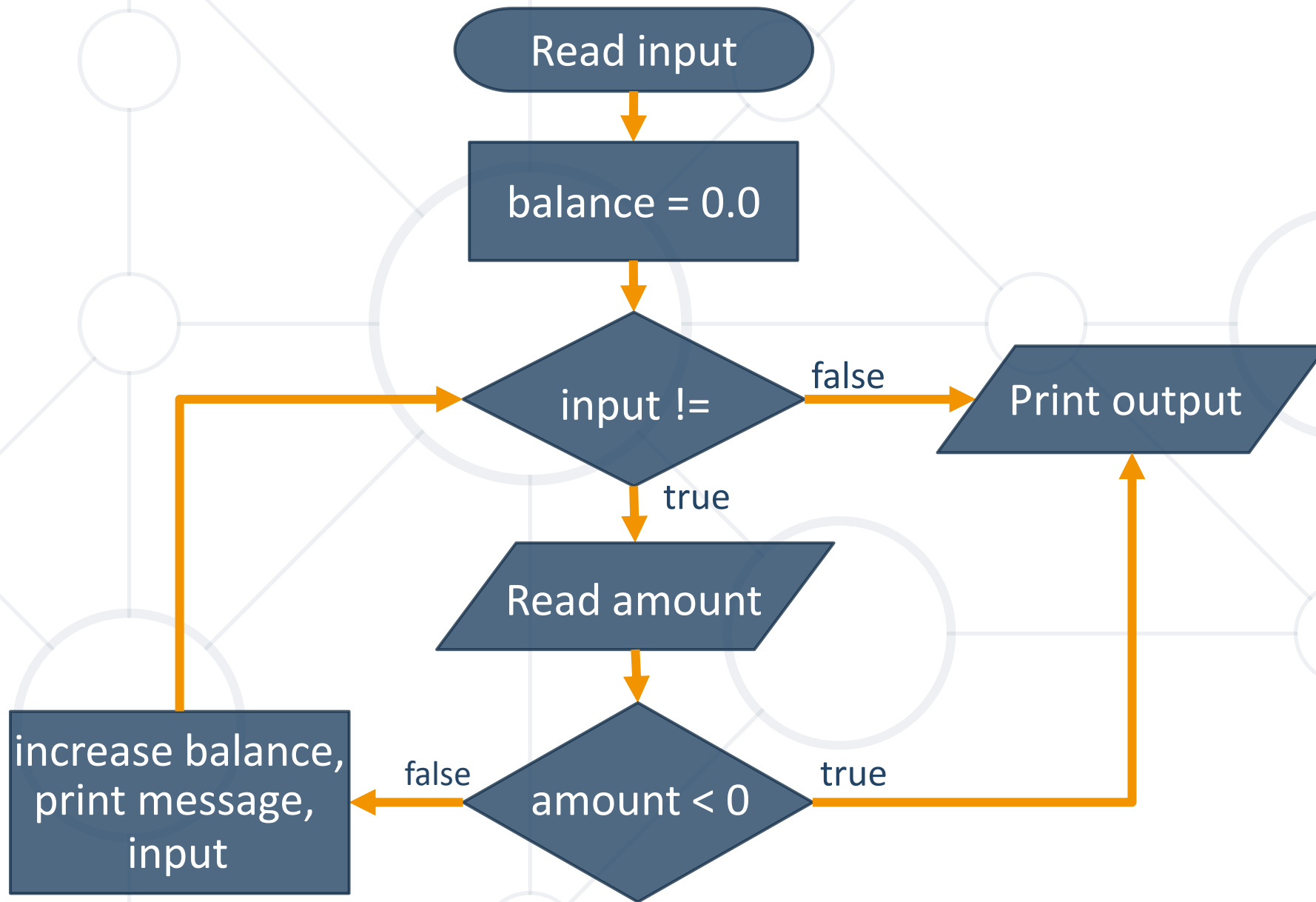


Increase: 5.51
Increase: 69.42
Increase: 100.00
Total: 174.93

120
45.55
-150



Increase: 120
Increase: 45.55
Invalid operation!
Total: 165.55



```
string input = Console.ReadLine();
double balance = 0.0;
while (input != "NoMoreMoney")
{
    double amount = double.Parse(input);
    if (amount < 0) { //TODO: Print output and exit the loop }
    balance += amount;
    Console.WriteLine($"Increase: {amount:F2}");
    input = Console.ReadLine();
}
Console.WriteLine($"Total: {balance:F2}");
```

Най-голямо число – условие

- Напишете програма, която:
 - Прочита **n** последователни пъти числа, докато получи команда "**Stop**"
 - Намира **най-голямото** измежду тях
- Примерен вход и изход:

100
99
80
70
Stop

 →

100

-10
20
-30
Stop

 →

20

45
-20
7
99
Stop

 →

99

5

3

Най-голямо число – решение

```
string input = Console.ReadLine();
int max = int.MinValue;
while (input != "Stop")
{
    int num = int.Parse(input);

    if (num > max)
    {
        max = num;
    }
    input = Console.ReadLine();
}
Console.WriteLine(max);
```

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/Practice/Index/3157#16>

Най-малко число – условие

- Напишете програма, която:
 - Прочита **n** последователни пъти числа, докато получи команда "**Stop**"
 - Намира **най-малкото** измежду тях
- Примерен вход и изход:

```
100  
99  
80  
70  
Stop
```



```
70
```

```
-10  
20  
-30  
Stop
```



```
-30
```

```
45  
-20  
7  
99  
Stop
```



```
-20
```



```
string input = Console.ReadLine();  
int min = int.MaxValue;  
while (input != "Stop")  
{  
    //TODO: Use logic similar  
    to the previous problem  
}
```

- Оператор **continue** – преминава към следващата итерация на цикъла

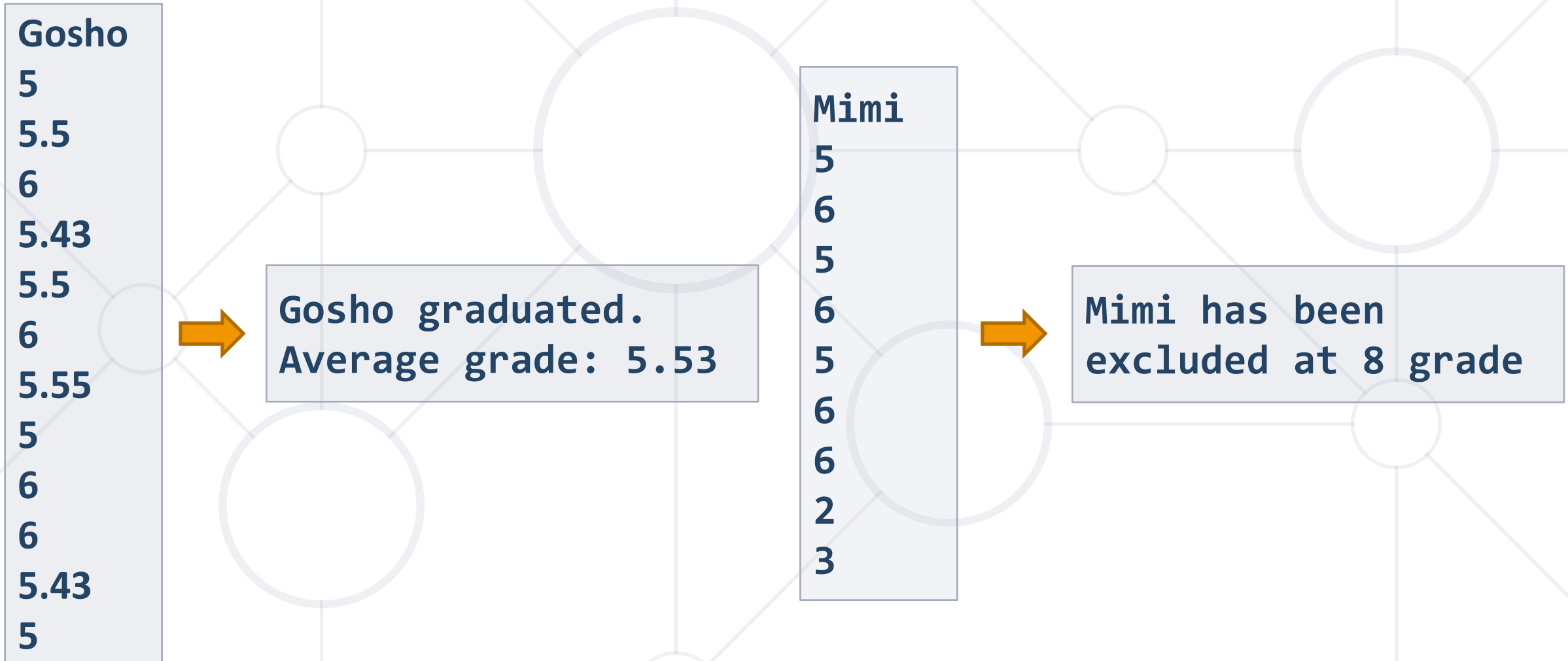
```
int i = 0;
while (i < 10)
{
    if (i % 2 == 0)
    {
        i++;
        continue;
    }
    Console.WriteLine(i);
    i++;
}
```



```
"C:\Program
1
3
5
7
9
```

- Напишете програма, която:
 - Изчислява **средната оценка** на ученик от цялото му обучение
 - Ако годишната му оценка е:
 - **≥ 4.00** , ученикът преминава е следващия клас
 - **< 4.00** , той ще повтори класа
 - Ако бъде скъсан повече от един път, той бива изключен и програмата приключва. Отпечатва се името и в кой клас е изключен:
"{име на ученика} has been excluded at {класа, в който е бил изключен} grade"
 - При **завършване** да се отпечата:
"{име на ученика} graduated. Average grade: {средната оценка от цялото обучение}"

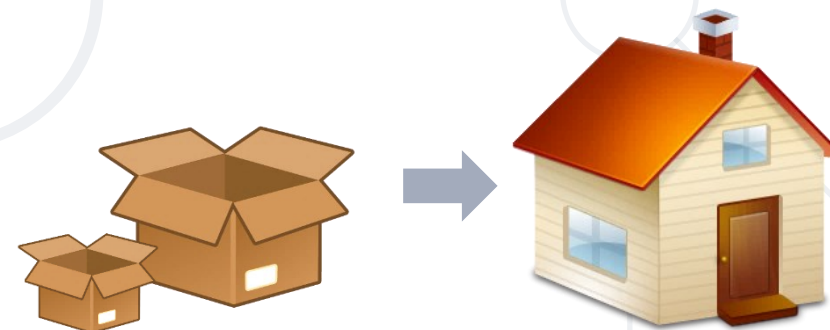
Завършване – условие (2)



```
string name = Console.ReadLine();
int grades = 1;
double sum = 0;
int excluded = 0;
while (grades <= 12)
{
    double grade = double.Parse(Console.ReadLine());
    if (grade < 4.00) {
        // TODO: increase excluded count and break if is more than 1
        continue;
    }
    // TODO: add grade to sum and increase grades count
}
double average = sum / 12; //TODO: print the output options
```



- Напишете програма, която:
 - Прочита 3 цели числа – ширина, дължина, височина
 - Прочита брой кашони до получаване на команда "Done"
 - Изчислява дали кашоните могат да се преместят в помещение с прочетените размери
 - 1 кашон е с размери 1м x 1м x 1м



- Ако помещението не може да събере кашоните, трябва да се принтира:
 - "No more free space! You need {брой недостигащи куб.метри} Cubic meters more."
- При получаване на команда "Done" и налично свободно място:
 - "{брой свободни куб. метри} Cubic meters left."

Преместване – условие (3)

- Примерен вход и изход:

10
1
2
4
6
Done



10 Cubic meters left.

2
2
2
9



No more free space! You need 1 Cubic meters more.



```
int width = int.Parse(Console.ReadLine());  
// TODO: Read the length and height  
  
int volume = width * length * height;  
bool hasVolume = true;  
  
string command = Console.ReadLine();  
while (!(command == "Done"))  
{  
    int box = int.Parse(command);  
    volume -= box;  
    ...  
}
```

Преместване – решение (2)

```
...
if (volume < 0)
{
    hasVolume = false;
    break;
}
command = Console.ReadLine();
}

if (hasVolume)
    Console.WriteLine("{0} Cubic meters left.", volume);
else
    Console.WriteLine("No more free space! You need {0}
    Cubic meters more.", Math.Abs(volume));
```

Цикълът прекъсва

Прочитаме отново командата



Вложени цикли

Пример – часовник (1)

Часовете се променят
когато минутите
надвишат 59

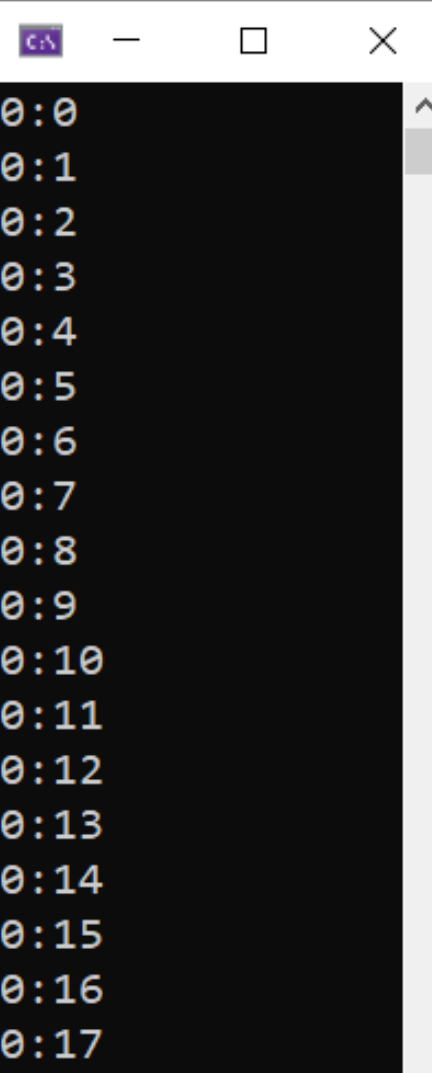
Докато минутите се
променят часовете
остават същите

19:03

Пример – часовник (2)

- Външният цикъл отговаря за часовете
 - Вътрешния за минутите

```
for (int h = 0; h <= 23; h++)  
{  
    for (int m = 0; m <= 59; m++)  
    {  
        Console.WriteLine($"{h}:{m}");  
    }  
}
```



```
0:0  
0:1  
0:2  
0:3  
0:4  
0:5  
0:6  
0:7  
0:8  
0:9  
0:10  
0:11  
0:12  
0:13  
0:14  
0:15  
0:16  
0:17
```

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

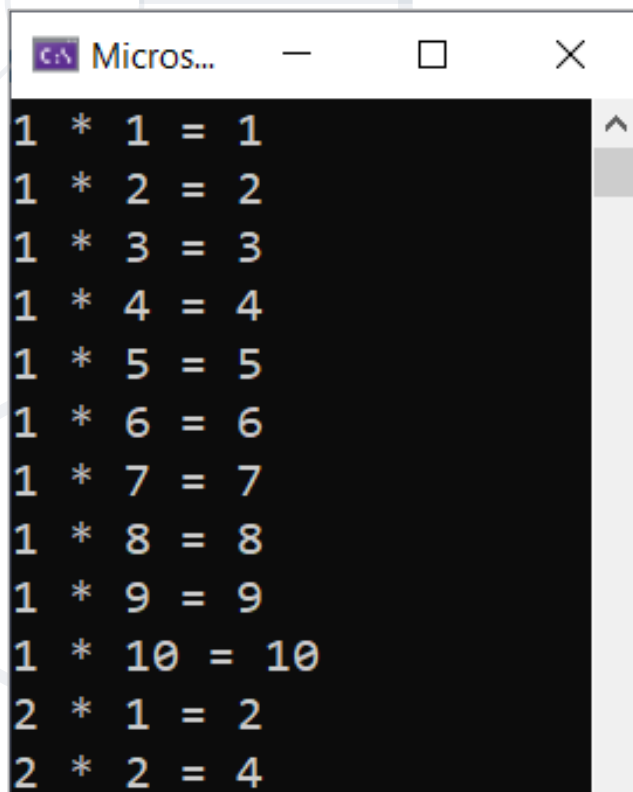
```
for (int i = 0; i < n; i++)  
    for (int j = 0; j < n; j++)  
        ...
```

Имената на
променливите трябва
да бъдат различни



Таблица за умножение – условие

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



```
C:\> Micros...  
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
```



Таблица за умножение – решение

```
for (int x = 1; x <= 10; x++)  
{  
    for (int y = 1; y <= 10; y++)  
    {  
        int product = x * y;  
        Console.WriteLine($"{x} * {y} = {product}");  
    }  
}
```

- За прекъсване на вложени цикли, използваме булеви променливи

```
bool flag = false;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (condition){
            flag = true;
            break;}
if (flag)
    break;
```

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

Сума от две числа – условие (1)

- 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**
 - Програмата **приключва изпълнение**
 - Ако **не се намери** нито една комбинация, 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



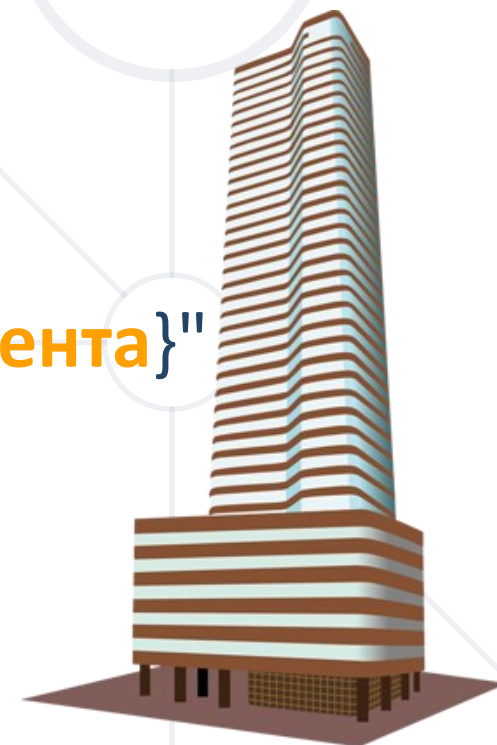
4 combinations - neither equals 20

Сума от две числа – решение

```
int startingNumber = int.Parse(Console.ReadLine());
int finalNumber = int.Parse(Console.ReadLine());
int magicNumber = int.Parse(Console.ReadLine());
int combinations = 0;
bool isFound = false;
for (int i = startingNumber; i <= finalNumber; i++)
    for (int j = startingNumber; j <= finalNumber; j++)
        combinations++;
        if (i + j == magicNumber)
            Console.WriteLine($"Combination N:{combinations} ({i} + {j}
                               = {magicNumber})");
            isFound = true;
            break;
if (isFound)
    break;
// Finish logic
```

Ако намерим комбинация, прекъсваме вътрешният цикъл

- 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**



- На последният етаж винаги има големи апартаменти
 - Те се означават с 'L', вместо с 'A'
- Ако има само един етаж, то има само **големи апартаменти**
- Входът се състои от **броя на етажите** и **броя на стаите** на един **етаж**
- Примерен вход и изход:

6
4



L60	L61	L62	L63
A50	A51	A52	A53
040	041	042	043
A30	A31	A32	A33
020	021	022	023
A10	A11	A12	A13

4
4



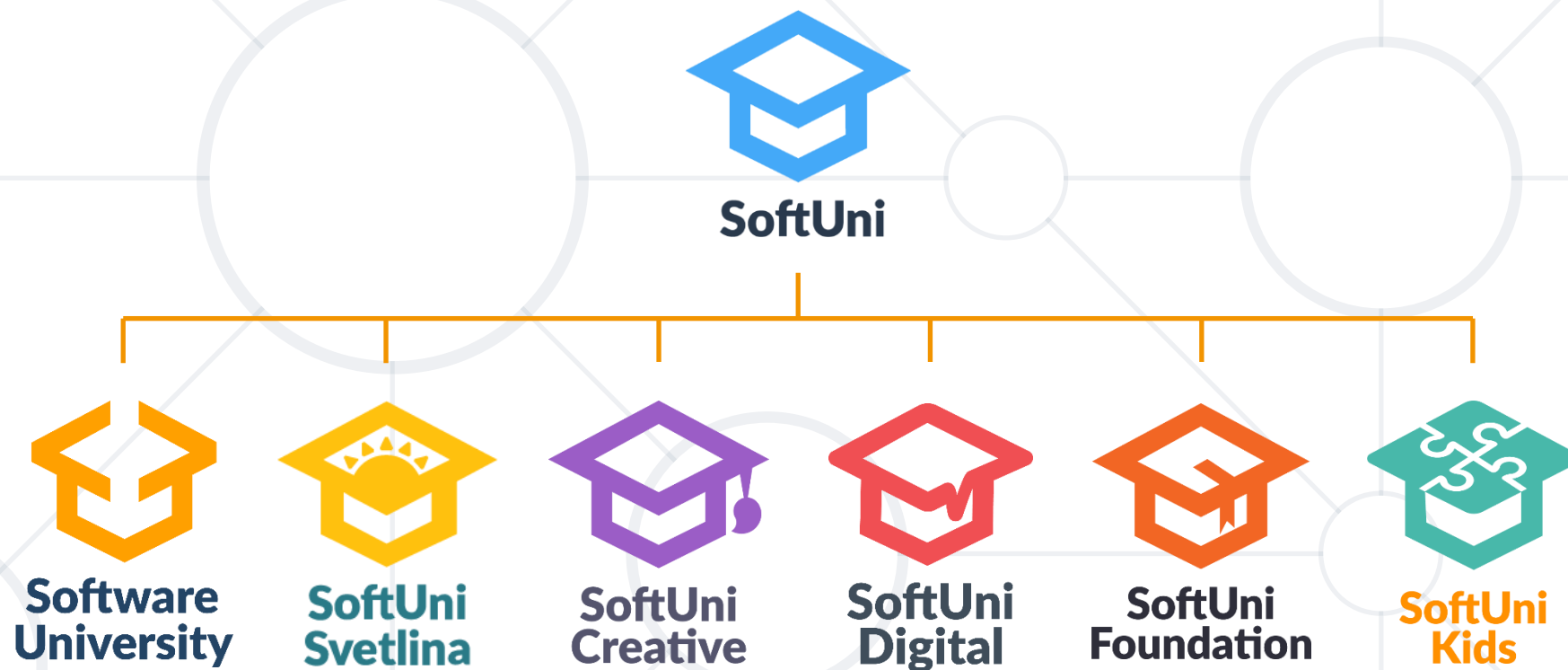
L40	L41	L42	L43
A30	A31	A32	A33
020	021	022	023
A10	A11	A12	A13

```
int floors = int.Parse(Console.ReadLine());
int rooms = int.Parse(Console.ReadLine());
for (int i = floors; i >= 1; i--)
{
    for (int j = 0; j < rooms; j++)
    {
        if (i == floors)
            Console.Write($"L{i}{j} ");
        // TODO: print according to floor number
    }
    Console.WriteLine();
}
```

Вложеният цикъл
итерира стаяте

- Повторение на блок код с **for**-цикъл
- Вземане на символ по индекс от текст
- Повторение на блок от код с **while**-цикъл
- Прекъсване на цикли с оператора **break**
- Преминаване към следваща итерация с оператора **continue**
- Вложени цикли

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
- Дискусионни форуми на СофтУни
 - forum.softuni.bg



Software University

