

Упражнения: Цикъл for

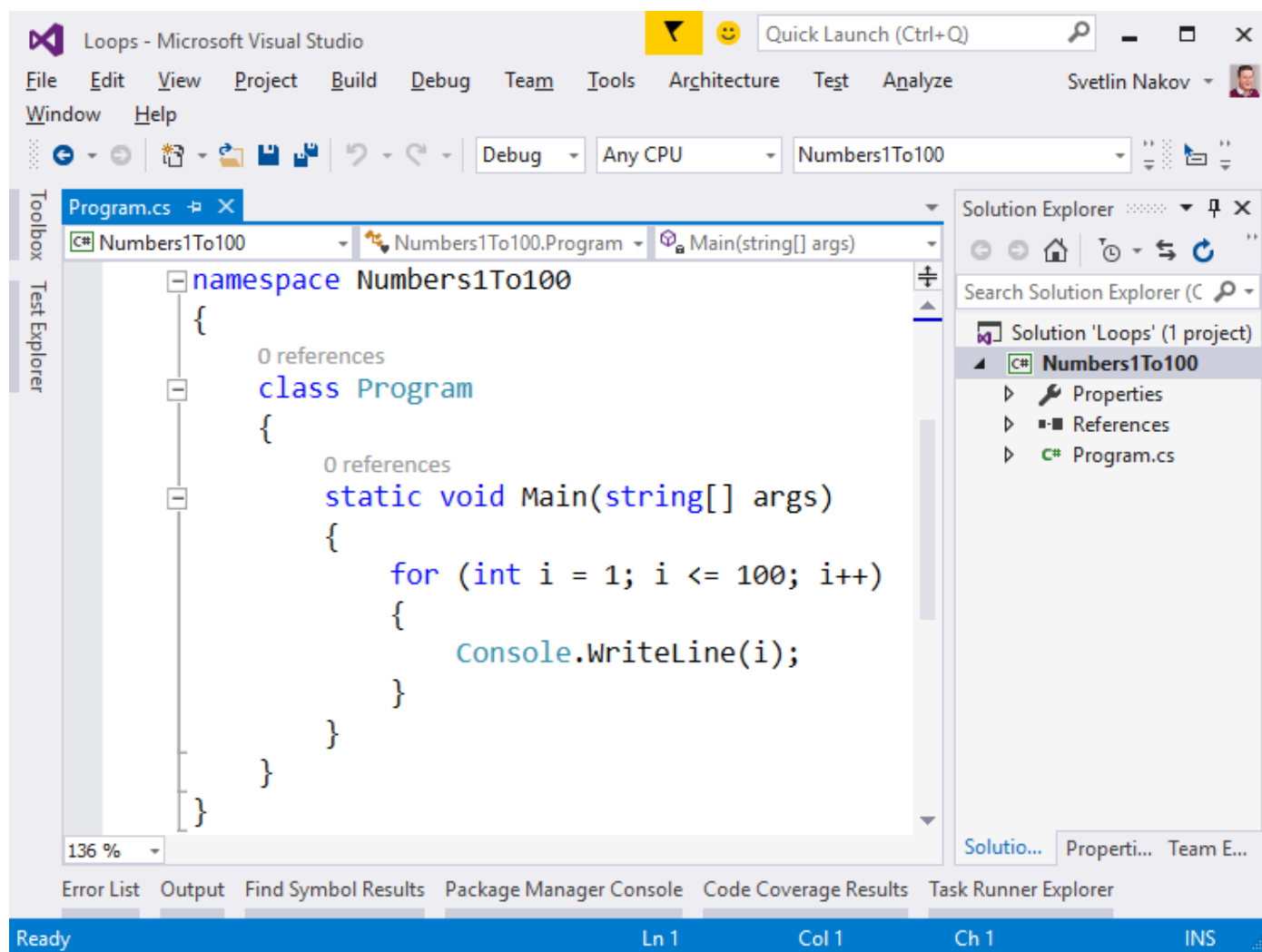
1. Числа от 1 до 100

Напишете програма, която отпечатва числата от 1 до 100, по едно на ред.

вход	изход
(няма)	1 2 3 ... 98 99 100

Подсказки:

1. Създайте **ново Visual Studio решение** за задачите от този урок и в него **нов проект** – конзолна C# програма. Задайте подходящо име на проекта, например „01.Числа от 1 до 100“ или „Numbers1To100“.
2. Отидете в тялото на метода **Main(string[] args)** и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



3. **Стартирайте** програмата с [Ctrl+F5] и я **тествайте**:

```

C:\WINDOWS\system32\cmd.exe
95
96
97
98
99
100
Press any key to continue . . .

```

2. Числа до 1000, завършващи на 7

Напишете програма, която отпечата числата в диапазона [1...1000], които завършват на 7.

вход	изход
(няма)	7 17 27 ... 997

Подсказка: можете да завъртите **for**-цикъл от 1 до 1000 и да проверите всяко число дали завършва на 7. Едно число **num** завършва на 7, когато **(num % 10 == 7)**.

3. Всички латински букви

Напишете програма, която отпечата всички букви от латинската азбука: **a, b, c, ..., z**.

Подсказка: можете да завъртите **for**-цикъл от 'a' до 'z' (освен числа може да въртите в цикъл и букви).

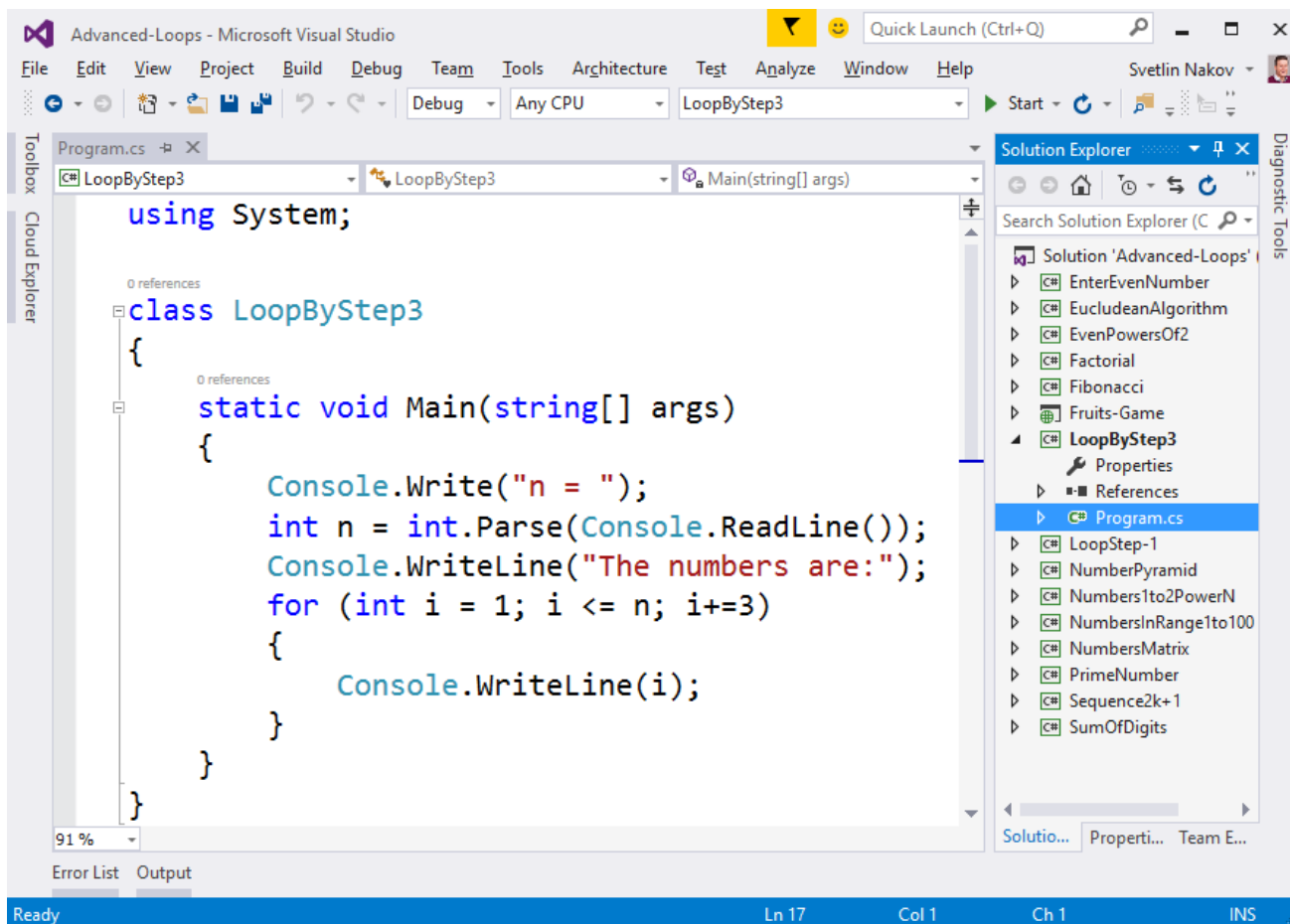
4. Числата от 1 до N през 3

Напишете програма, която чете число **n**, въведено от потребителя, и отпечата числата от 1 до **n** през 3 (със стъпка 3). Примери:

вход	изход	вход	изход	вход	изход
10	1 4 7 10	7	1 4 7	15	1 4 7 10 13

Подсказки:

1. Можете да завъртите **for**-цикъл със стъпка 3 по следния начин: **for (var i = 1; i <= n; i+=3)**.
2. Отидете в тялото на метода **Main(string[] args)** и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



3. **Стартирайте** програмата с [Ctrl+F5] и я **тествайте**:

```

C:\WINDOWS\system32\cmd.exe

n = 10
The numbers are:
1
4
7
10
Press any key to continue . . .

```

```

C:\WINDOWS\system32\cmd.exe

n = 8
The numbers are:
1
4
7
Press any key to continue . . .

```

5. Числата от N до 1 в обратен ред

Напишете програма, която чете цяло положително число **n**, въведено от потребителя, и печата **числата от n до 1 в обратен ред** (от най-голямото към най-малкото). Примери:

вход	изход	вход	изход	вход	изход
2	2 1	3	3 2 1	5	5 4 3 2 1

Подсказка: отпечатайте **n** звездички в цикъл **n** пъти, точно както в предната задача.

6. Числа – степени на 2

Напишете програма, която чете от конзолата цяло число **n**, въведено от потребителя, и **печата числата от 1 до 2ⁿ**. Примери:

вход	изход
3	1 2 4 8

вход	изход
4	1 2 4 8 16

вход	изход
5	1 2 4 8 16 32

Подсказка: завъртете **for**-цикъл от **0** до **n** и започвайки от **num = 1** на всяка стъпка умножавайте **num** по **2**.

7. Четни степени на 2

Да се напише програма, която чете число **n**, въведено от потребителя, и **печата четните степени на 2** $2 \leq 2^n$: **2⁰, 2², 2⁴, 2⁸, ..., 2ⁿ**. Примери:

вход	изход
3	1 4

вход	изход
4	1 4 16

вход	изход
5	1 4 16

вход	изход
6	1 4 16 64

вход	изход
7	1 4 16 64

Подсказка: започнете от 1 и в цикъл умножавайте по 4 на всяка стъпка.

8. Сумиране на числа

Да се напише програма, която **чете n**-на брой **цели числа**, въведени от потребителя, и **ги сумира**.

- От първия ред на входа се въвежда броят числа **n**.
- От следващите **n** реда се въвежда по едно цяло число.

Програмата трябва да прочете числата, да ги сумира и да отпечата сумата им. Примери:

вход	изход
2 10 20	30

вход	изход
3 -10 -20 -30	-60

вход	изход
4 45 -20 7 11	43

вход	изход
1 999	999

вход	изход
0	0

Подсказки:

- Първо въведете едно число **n** (броят числа, които предстои да бъдат въведени).
- Инициализирайте **sum = 0** (в началото няма още прочетени числа, и съответно сумата е празна).
- В цикъл **n** пъти прочетете по едно цяло число **num** и го прибавете към сумата (**sum = sum + num**).
- Накрая в **sum** трябва да се е запазила сумата на прочетените числа. Отпечатайте я.

9. Най-голямо число

Напишете програма, която чете **n-на брой цели числа** ($n > 0$), въведени от потребителя, и намира **най-голямото** измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа, по едно на ред.

Примери:

вход	изход	вход	изход	вход	изход	вход	изход	вход	изход
2	100	3	20	4	99	1	999	2	-1
100		-10		45		999		-1	
99		20		-20				-2	
		-30		7					
				99					

Подсказки:

- Първо въведете едно число **n** (брой числа, които предстои да бъдат въведени).
- Въведете от конзолата първото число. Сложете текущият максимум **max** да е прочетеното число.
- В цикъл **n-1 пъти** прочетете по едно цяло число **num**. Ако прочетеното число **num** е по-голямо от текущият максимум **max**, запомнете **num** в **max**.
- Накрая в **max** трябва да се е запазило най-голямото число. Отпечатайте го.

10. Най-малко число

Напишете програма, която чете **n-на брой цели числа** ($n > 0$), въведени от потребителя, и намира **най-малкото** измежду тях. Първо се въвежда броят числа **n**, а след това самите **n** числа, по едно на ред. Примери:

вход	изход	вход	изход	вход	изход	вход	изход	вход	изход
2	99	3	-30	4	-20	1	999	2	-2
100		-10		45		999		-1	
99		20		-20				-2	
		-30		7					
				99					

Подсказки: задачата е абсолютно аналогична с предходната.

11. Лева и дясна сума

Да се напише програма, която чете **2*n-на брой цели числа**, подадени от потребителя, и проверява дали **сумата на първите n числа** (лева сума) е равна на **сумата на вторите n числа** (дясна сума). При равенство печата "Yes" + **сумата**; иначе печата "No" + **разликата**. Разликата се изчислява като положително число (по абсолютна стойност). Примери:

вход	изход	коментар	вход	изход	коментар
2	Yes, sum = 100	$10+90 = 60+40 = 100$	2	No, diff = 1	$90+9 \neq 50+50$
10			90		Difference =
90			9		$ 99-100 = 1$
60			50		
40			50		

Подсказки:

- Въведете **n**.
- Въведете първите **n** числа (**лявата** половина) и ги сумирайте.

- Въведете още **n** числа (**дясната** половина) и ги сумирайте.
- Изчислете **разликата** между сумите по абсолютна стойност: `Math.Abs(leftSum - rightSum)`.
- Ако разликата е **0**, отпечатайте **"Yes" + сумата**; иначе отпечатайте **"No" + разликата**.

12. Четна / нечетна сума

Да се напише програма, която чете **n-на брой** цели числа, подадени от потребителя, и проверява дали **сумата от числата на четни позиции** е равна на **сумата на числата на нечетни позиции**. При равенство да се отпечата **"Yes" + сумата**; иначе да се отпечата **"No" + разликата**. Разликата се изчислява по абсолютна стойност. Примери:

вход	изход	коментар
4 10 50 60 20	Yes Sum = 70	10+60 = 50+20 = 70

вход	изход	коментар
4 3 5 1 -2	No Diff = 1	3+1 ≠ 5-2 Diff = 4-3 = 1

вход	изход	коментар
3 5 8 1	No Diff = 2	5+1 ≠ 8 Diff = 6-8 = 2

Подсказки: Въведете числата едно по едно и изчислете двете **суми** (числа на **четни** позиции и числа на **нечетни** позиции). Както в предходната задача, изчислете абсолютна стойност на разликата и отпечатайте резултата (**"Yes" + сумата** при разлика 0 или **"No" + разликата** в противен случай).

13. Еднакви двойки

Дадени са **2*n-на брой** числа. Първото и второто формират **двойка**, третото и четвъртото също и т.н. Всяка двойка има **стойност** – сумата от съставлящите я числа. Напишете програма, която проверява **дали всички двойки имат еднаква стойност** или печата **максималната разлика** между две последователни двойки. Ако всички двойки имат еднаква стойност, отпечатайте **"Yes, value={Value}" + стойността**. В противен случай отпечатайте **"No, maxdiff={Difference}" + максималната разлика**. Примери:

вход	изход	коментари
3 1 2 0 3 4 -1	Yes, value=3	стойности = {3, 3, 3} еднакви стойности
4 1 1 3 1 2 2 0 0	No, maxdiff=4	стойности = {2, 4, 4, 0} разлики = {2, 0, 4} макс. разлика = 4
2 -1 0	Yes, value=-1	стойности = {-1, -1} еднакви стойности

вход	изход	коментари
2 1 2 2 2	No, maxdiff=1	стойности = {3, 4} разлики = {1} макс. разлика = 1
1 5 5	Yes, value=10	стойности = {10} една стойност еднакви стойности
2 -1 2	No, maxdiff=2	стойности = {1, -1} разлики = {2} макс. разлика = 2

0 -1			0 -1		
---------	--	--	---------	--	--

Подсказки:

- Прочитайте входните числа **по двойки**. За всяка двойка пресмятайте **сумата**.
- Докато четете входните двойки, за всяка двойка без първата пресмятайте **разликата с предходната**. За целта пазете в отделна променлива сумата на предходната двойка.
- Намерете **най-голямата разлика** между две двойки. Ако е **0**, печатайте **"Yes"** иначе **"No"** + разликата.

14. * Елемент, равен на сумата на останалите

Да се напише програма, която чете **n-на брой** цели числа, въведени от потребителя, и проверява дали сред тях съществува число, което е равно на сумата на всички останали. Ако има такъв елемент, печата **"Yes"**, **"Sum = "** + **неговата стойност**; иначе печата **"No"**, **"Diff = "** + **разликата между най-големия елемент и сумата на останалите** (по абсолютна стойност).

Примери:

вход	изход	коментари
7 3 4 1 1 2 12 1	Yes Sum = 12	$3 + 4 + 1 + 2 + 1 + 1 = 12$
4 6 1 2 3	Yes Sum = 6	$1 + 2 + 3 = 12$
3 1 1 10	No Diff = 8	$ 10 - (1 + 1) = 8$
3 5 5 1	No Diff = 1	$ 5 - (5 + 1) = 1$
3 1 1 1	No Diff = 1	

Подсказка: изчислете **сумата** на всички елементи и **най-големият** от тях и проверете търсеното условие.