

Упражнения: Повторения с различни стъпки

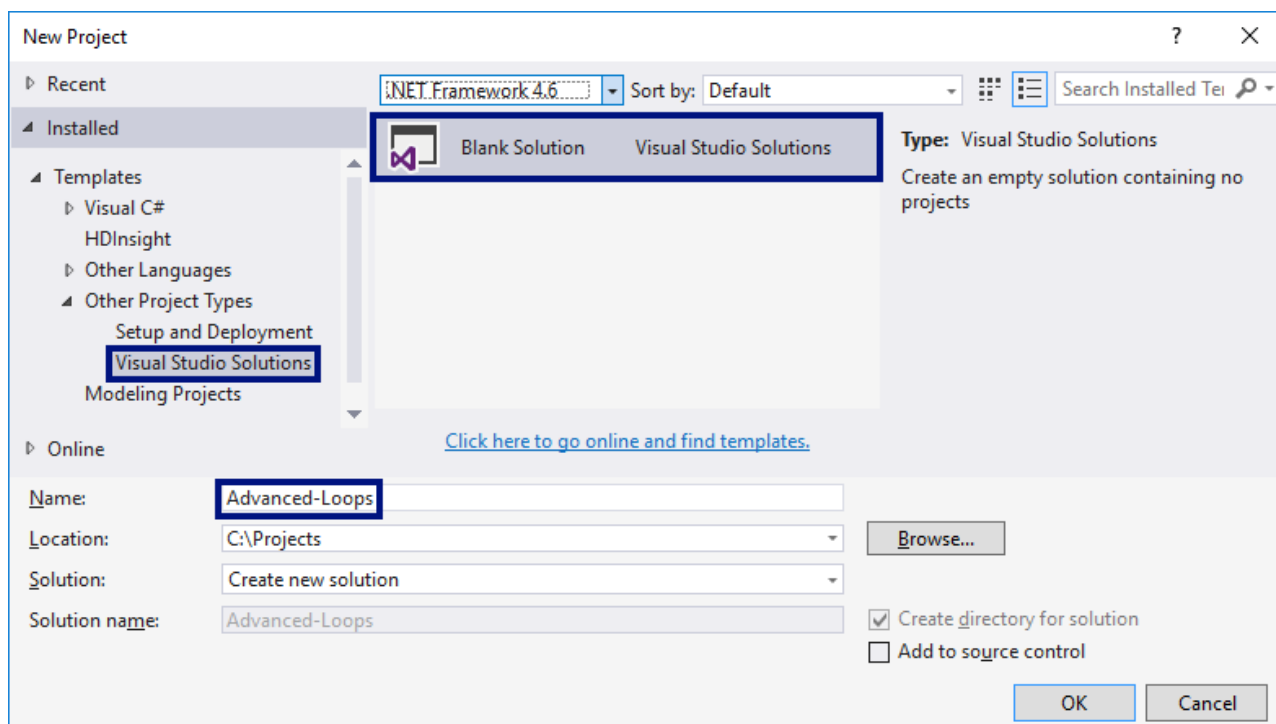
Практически упражнения към курса "[Увод в програмирането](#)" за ученици.

Тествайте решенията си от тази тема в Judge:

<https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>

0. Празно Visual Studio решение (Blank Solution)

1. Създайте празно решение (**Blank Solution**) във Visual Studio за да организирате кода от задачите за упражнение. Целта на този **blank solution** е да съдържа **по един проект за всяка задача** от упражненията.



2. Задайте **да се стартира по подразбиране текущия проект** (не първият в решението). Кликнете с десен бутон на мишката върху **Solution 'Advanced-Loops' → [Set StartUp Projects...] → [Current selection]**.

1. Числата от 1 до N през 3

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

ВХОД	ИЗХОД
10	1 4 7 10

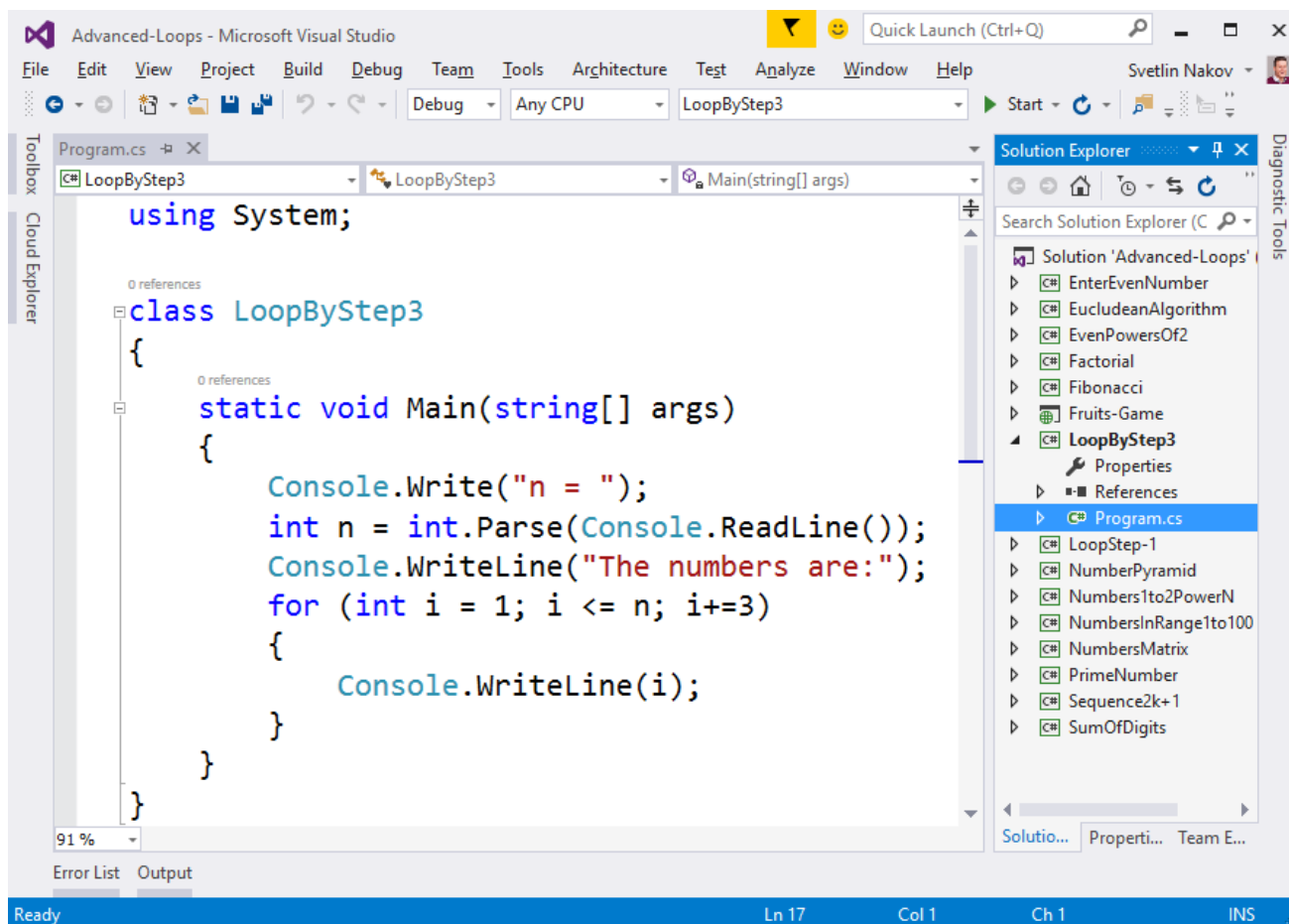
ВХОД	ИЗХОД
7	1 4 7

ВХОД	ИЗХОД
15	1 4 7 10 13

Подсказки:

1. Създайте **нов проект** в съществуващото Visual Studio решение – конзолна C# програма. Задайте подходящо име на проекта, например **“LoopByStep3”**.
2. Можете да завъртите **for-цикъл със стъпка 3** по следния начин: **for (var i = 1; i <= n; i+=3)**.

3. Отидете в тялото на метода **Main(string[] args)** и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



4. **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. **Проверете** решението си в **judge системата**.

Отворете страницата в judge за този урок: <https://judge.softuni.bg/Contests/2640/Повторения-с-различни-стъпки>.

Изберете задачата “**Числата от 1 до N през 3**”. Копирайте и поставете в тъмното поле **сурс кода**. Натиснете бутона за изпращане [Submit]:

→ ↻ 🏠 judge.softuni.bg/Contests/Practice/Index/2640#0

Results

Submit a solution

Числата от 1 до N през 3
Числата от N до 1 в обратен ред
Числа – степени на 2
Четни степени на 2

Редица числа 2K+1
Число в диапазона от 1 до 100
Най-голям общ делител (НОД)

Числата от 1 до N през 3

🔗 Условия |

```

1 using System;
2
3 class LoopByStep3
4 {
5     //1.Числата от 1 до N през 3
6     static void Main()
7     {
8         var n = int.Parse(Console.ReadLine());
9         for (int k = 1; k <= n; k += 3)
10        {
11            Console.WriteLine(k);
12        }
13    }
14 }

```

Allowed working time: 0.100 sec.
 Allowed memory: 16.00 MB
 Size limit: 16.00 KB
 Checker: Trim

C# code Submit

Трябва да получите **100 точки** (напълно вярна задача):

⏮ ⏪ 1 ⏩ ⏭			🔄
Points	Time and memory used	Submission date	
✓✓✓✓ 100 / 100	Memory: 7.40 MB Time: 0.031 s	11:27:09 05.11.2020	Details

2. 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.

3. 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**.

4. Четни степени на 2

Да се напише програма, която чете число **n**, въведено от потребителя, и **печата четните степени на 2** $2 \leq 2^n$: $2^0, 2^2, 2^4, 2^8, \dots, 2^n$. 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.

5. Редица числа $2K+1$

Напишете програма, която чете число **n**, подадено от потребителя, и отпечата **всички числа** $\leq n$ от **редицата**: 1, 3, 7, 15, 31, Всяко следващо число се изчислява като **предишното число * 2 + 1**. Примери:

вход	изход
3	1 3

вход	изход
8	1 3 7

вход	изход
17	1 3 7 15

вход	изход
31	1 3 7 15 31

Подсказки:

- Започнете от **num = 1**.
- В цикъл докато **num** не стигне **n**, печатайте, го умножавайте по **2** и прибавяйте **1**.

6. Число в диапазона от 1 до 100

Напишете програма, която чете цяло положително **число n** в **диапазона [1...100]**, въведено от потребителя. При въвеждане на число извън посочения диапазон, да се отпечата съобщение за грешка и потребителят да се подкани **да въведе ново число**. Примери:

вход / изход
35 The number is: 35
105 Invalid number!
0 Invalid number!

```
-200
Invalid number!
77
The number is: 77
```

Подсказки:

- Въведете число.
- Повтаряйте в цикъл докато числото е невалидно: отпечатайте грешка и въведете число отново.

7. Най-голям общ делител (НОД)

Напишете програмата, която чете две цели положителни числа **a** и **b**, въведени от потребителя, и изчислява и отпечатва **най-големият им общ делител (НОД)**. Примери:

вход	изход
24 16	8

вход	изход
67 18	1

вход	изход
15 9	3

вход	изход
100 88	4

вход	изход
10 10	10

Подсказка: имплементирайте **алгоритъма на Евклид**: <https://bg.wikipedia.org/wiki/алгоритъм-на-Евклид>.

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма **"Обучение за ИТ кариера"** на МОН за подготовка по професия "Приложен програмист".



- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

