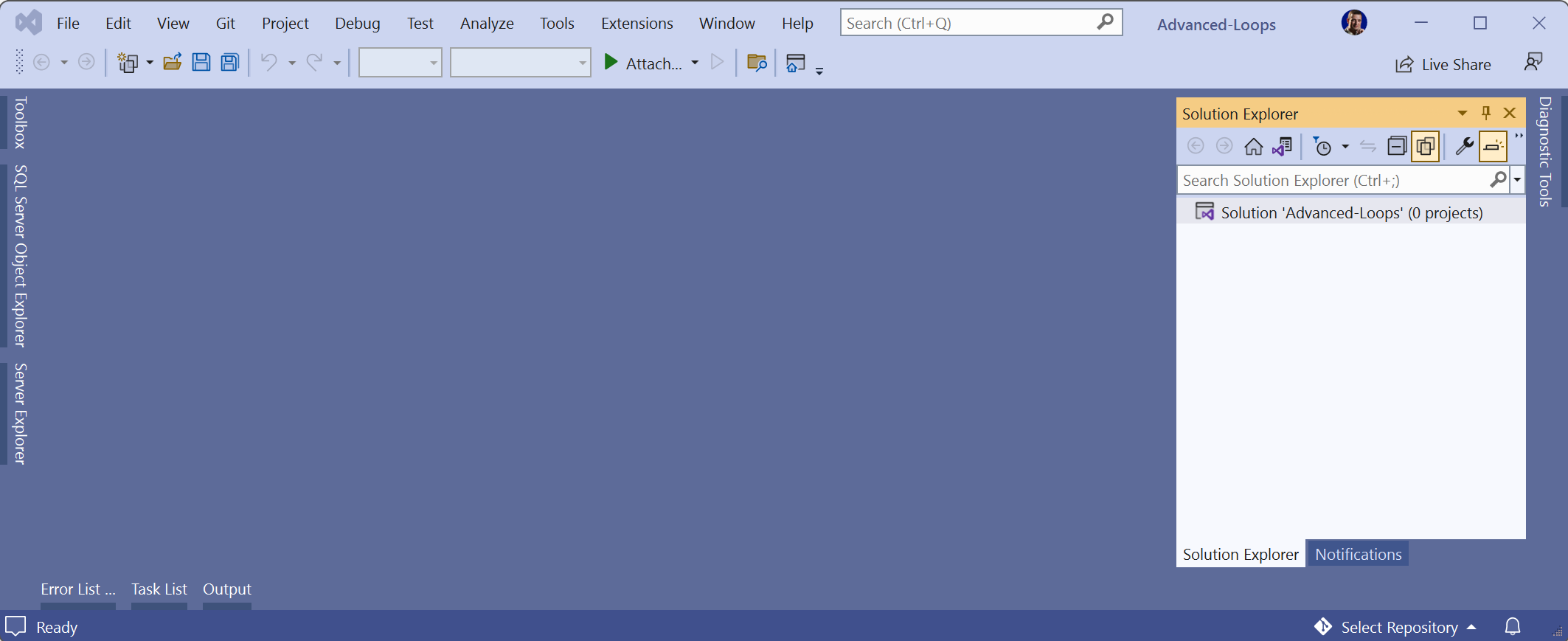
# Упражнения: Повторения с различни стъпки

## Празно решение

Създайте празно решение, за да организирате задачите от упражненията.

1. Стартирайте Visual Studio.
2. Създайте нов проект [Create New Project].
3. Търсете и изберете: Blank Solution
4. Именувайте решението: Advanced-Loops

Сега имате създадено празно решение без проекти в него:



Добавяйте в решението по един проект за всяка задача от упражненията. Така всяка задача ще бъде в отделен проект и всички проекти ще бъдат в общото решение.

Не забравяйте да зададете да се стартира по подразбиране текущия проект, a не първият в решението, като кликнете с десен бутон на мишката върху Solution 'Advanced-Loops' 🡪 [Set StartUp Projects…] 🡪 [Current selection].

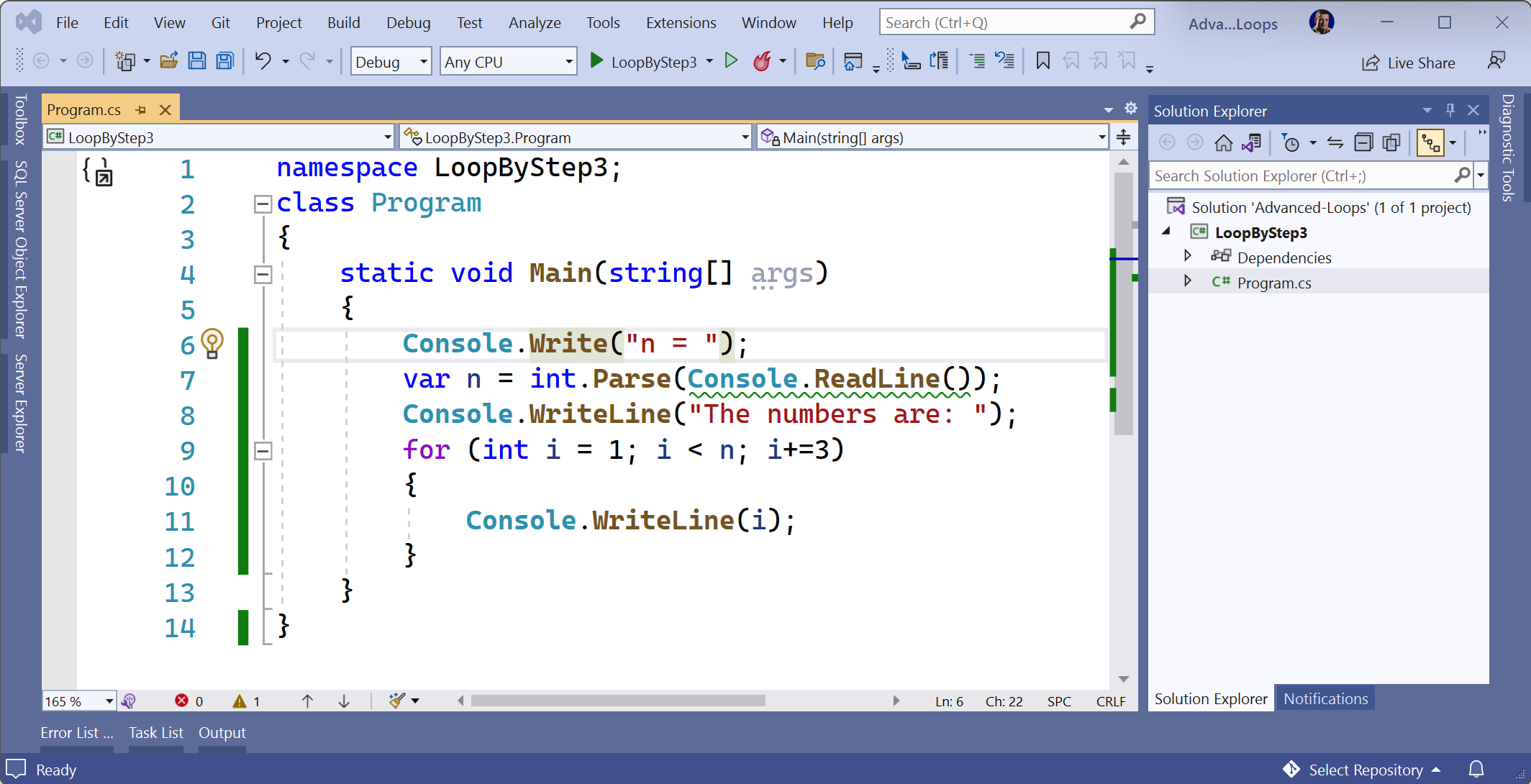
## Числата от 1 до N през 3

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

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

Подсказки:

1. Създайте нов проект от менюто [File] 🡪 [New] 🡪 [New Project …]
2. Изберете шаблон на проекта [C#] 🡪 [Windows] 🡪 [Console] 🡪 [Console App].
3. Именувайте проекта: LoopByStep3.
4. Напишете кода на програмата:



1. Стартирайте с [Ctrl+F5] и тествайте:

Text

Description automatically generated

## 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.

## 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 |  | вход | изход |  | вход | изход |
| 3 | 1  2  4  8 | 4 | 1  2  4  8  16 | 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.

## Четни степени на 2

Да се напише програма, която чете число n, въведено от потребителя, и печата четните степени на 2 ≤ 2n: 20, 22, 24, 28, …, 2n. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 3 | 1  4 | 4 | 1  4  16 | 5 | 1  4  16 | 6 | 1  4  16  64 | 7 | 1  4  16  64 |

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

## Редица числа 2K+1

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

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 3 | 1  3 | 8 | 1  3  7 | 17 | 1  3  7  15 | 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.

## 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 |
| Еnter a number in the range [1...100]: 35  The number is: 35 |
| Еnter a number in the range [1...100]: 105  Invalid number!  Еnter a number in the range [1...100]: 0  Invalid number!  Еnter a number in the range [1...100]: -200  Invalid number!  Еnter a number in the range [1...100]: 77  The number is: 77 |

Подсказки:

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

## Най-голям общ делител (НОД)

Напишете програма, 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 |  | вход | изход |  | вход | изход |  | вход | изход |  | вход | изход |
| 24  16 | 8 | 67  18 | 1 | 15  9 | 3 | 100  88 | 4 | 10  10 | 10 |

Подсказка: имплементирайте алгоритъма на Евклид: [https://bg.wikipedia.org/wiki/алгоритъм-на-Евклид](https://bg.wikipedia.org/wiki/Алгоритъм_на_Евклид).