

# Упражнения: Повторения от по-висока сложност

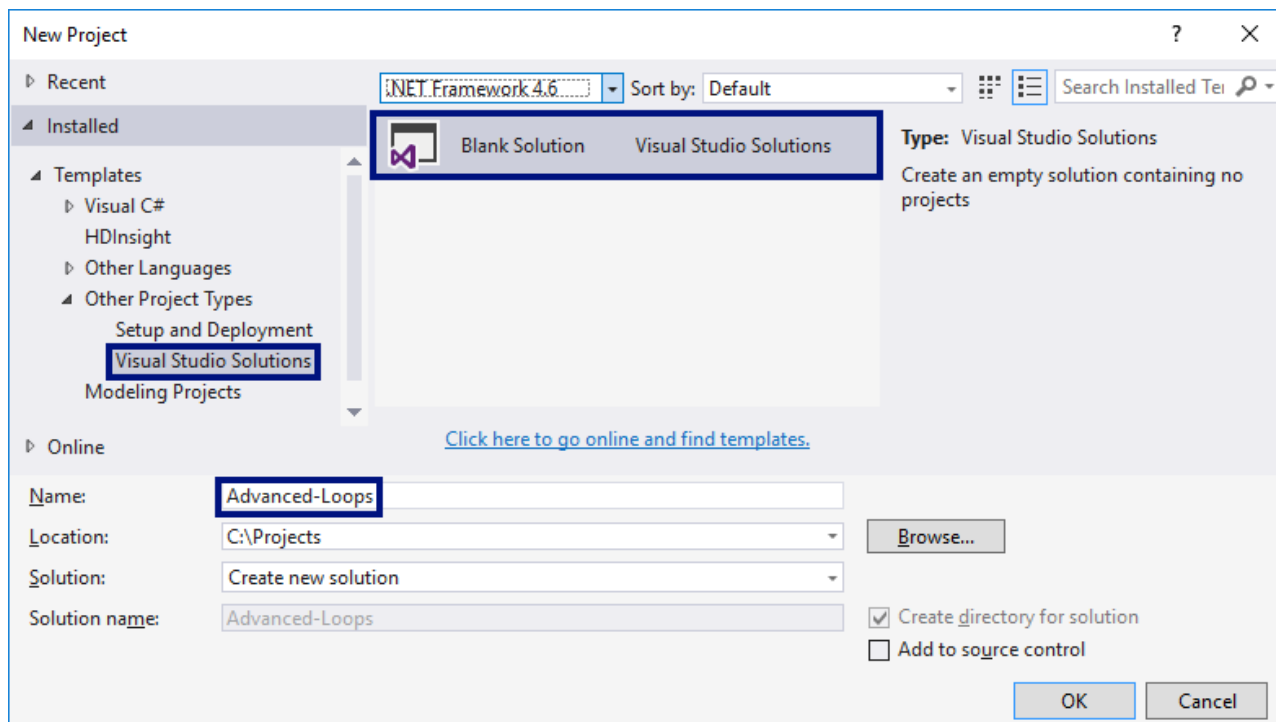
Практически упражнения към курса "[Увод в програмирането](#)" за ученици.

Тествайте решенията си от тази тема в Judge:

<https://judge.softuni.bg/Contests/2641/Повторения-от-по-висока-сложност>

## 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. Факториел

Напишете програма, която чете цяло число  $n$  ( $1 \leq n \leq 12$ ), въведено от потребителя, и **изчислява и отпечатва**  $n! = 1 * 2 * \dots * n$  (**n факториел**). Примери:

ВХОД	ИЗХОД
5	120

ВХОД	ИЗХОД
6	720

ВХОД	ИЗХОД
10	3628800

ВХОД	ИЗХОД
1	1

ВХОД	ИЗХОД
2	2

**Подсказка:** в цикъл умножете числата от 1 до  $n$ .

## 2. Сумиране на цифрите на число

Напишете програма, която чете цяло число **num**, въведено от потребителя, и отпечатва **сумата от цифрите му**.

Примери:

ВХОД	ИЗХОД	коментар
------	-------	----------

ВХОД	ИЗХОД	коментар
------	-------	----------

ВХОД	ИЗХОД
------	-------

ВХОД	ИЗХОД
------	-------

5634	18	6+6+3+4 = 18	19	10	1+9 = 10	5	5	17151	15
------	----	--------------	----	----	----------	---	---	-------	----

**Подсказка:** в цикъл докато не стигнете до 0 сумирайте последната цифра на числото (`num % 10`) и го разделяйте след това на 10 (така изтривате последната му цифра).

### 3. Проверка за просто число

Напишете програма, която чете цяло число **n**, въведено от потребителя, и **проверява дали е просто число** (дали се дели само на себе си и на единица). Да се отпечата **“Prime”** или **“Not prime”**. Примери:

вход	изход	вход	изход	вход	изход	вход	изход	вход	изход
2	Prime	3	Prime	4	Not Prime	5	Prime	7	Prime

вход	изход	вход	изход	вход	изход	вход	изход
1	Not Prime	0	Not Prime	-1	Not Prime	149	Prime

**Подсказки:**

- Ако числото е по-малко от 2, значи не е просто.
- Всяко друго число първоначално се приема за **просто** и се проверява в цикъл дали се дели на числата 2, 3, 4, ..., до корен квадратен от **n** (цялата част).

### 4. Въвеждане на четно число (с обработка на грешен вход)

Напишете програма, която **чете четно число**, въведено от потребителя. Ако потребителят въведе **грешно число** (нечетно число, стринг или число, което не е цяло), трябва да му излиза **съобщение за грешка** и да **въвежда отново**. Примери:

вход / изход
34 Even number entered: 34
35 The number is not even. hello Invalid number! 12.85 Invalid number! 3464232636536513 Invalid number! 8 Even number entered: 8

**Подсказки:**

- В цикъл въвеждайте число и проверявайте дали е четно. При коректно число излезте от цикъла.
- С `try { ... } catch { ... }` конструкция прихванете грешните числа, които не могат да се обърнат в `int`.

### 5. Числа на Фибоначи

Напишете програма, която чете цяло число **n**, въведено от потребителя, и пресмята **n-тото число на Фибоначи**. Нулевото число на Фибоначи е 1, първото е също 1, а всяко следващо е сумата от предходните две. Примери:

вход	изход
0	1

вход	изход
1	1

вход	изход
2	2

вход	изход
5	8

вход	изход
10	89

Подсказка:

- При  $n < 2$  отпечатайте 1.
- Започнете от  $f0=1$  и  $f1=1$  и в цикъл сумирайте последните две числа. Записвайте последните две числа след всяка стъпка в  $f0$  и  $f1$ .

## 6. Пирамида от числа

Напишете програма, която чете цяло число  $n$ , въведено от потребителя, и отпечата **пирамида от числа** като в примерите:

вход	изход
7	1 2 3 4 5 6 7

вход	изход
10	1 2 3 4 5 6 7 8 9 10

вход	изход
12	1 2 3 4 5 6 7 8 9 10 11 12

вход	изход
15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Подсказка:

- С **два вложени цикъла** печатайте пирамида от числа: на първия ред едно число, на втория ред 2 числа, на третия ред 3 числа и т.н.
- В отделен **брояч** пазете колко числа сте отпечатали до момента (и кое е текущото число). Когато стигнете  $n$ , излезте внимателно от двата вложени цикъла с **break** или **return**.

## 7. Таблица с числа

Напишете програма, която чете цяло число  $n$ , въведено от потребителя, и отпечата **таблица (матрица) от числа** като в примерите:

вход	изход
2	1 2 2 1

вход	изход
3	1 2 3 2 3 2 3 2 1

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

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

Подсказка:

- С **два вложени цикъла** за **row** (ред) и **col** (колона) печатайте число по формулата  $num = row + col + 1$ .
- За долната дясна половина на таблицата ще се получат грешни резултати. Там използвайте формулата  $2*n - num$ .
- Как се сетихме за тези формули? Математическа досетливост: наблюдаваме числата, предполагаме каква е формулата, тестваме и ако не се получи, измисляме друга формула и пак пробваме. В случая имаме различни формули за горната лява и долната дясна половина от матрицата.

## 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".



Министерство  
на образованието  
и науката



Национална  
програма  
„Обучение за  
ИТ кариера“

- 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).



SoftUni  
Foundation

